1994年6月1日発行（毎月1回1日発行）第13巻6号通巻146号 昭和58年11月2日第三種郵便物認可

特集 X68000と仲間たち

040turbo/POLYPHON-24/GF-540q/SP-120MO/満開式磁帯駆動装置壱號
DrawingSlate/PPIボード/無停電電源装置BX3/メガディスプレイへの道

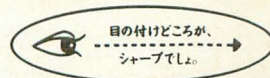
第5回アンケート分析大会/マイクロコンピュータショウ'94レポート

6
1994

SOFT BANK オー／エックス
定価600円

ビデオグラフィックスの
世界へ。
資料請求券
X68030
Oh!X
6除
■お問い合わせは…シャープ株式会社
電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

SHARP

目の付けどころが、シャープでしょ。

1,677万色対応、ビデオ映像を高画質・高速取り込み

テレビやビデオ、ビデオディスクなどの映像をX68シリーズやMacシリーズ※1の動画・静止画データとして高速取り込みが可能、いわば“ビデオスキャナ”とでも呼びたいビデオ入力ユニットです。1,677万色対応、最大640×480ドットの高解像度※2。動画・静止画の手軽なハンドリングが、新たなグラフィックシーンを創造します。

※1 MacintoshはIIシリーズ以降の機種に対応、ディスプレイ解像度が640×480ドットの場合、取り込み可能な範囲は、160×120ドット、320×240ドットのサイズになります。
※2 X68030/X68000シリーズでは、1,677万色はデータ作成のみに対応。表示は最大65,536色、解像度は512×512ドット。また、Macintoshは機種により表示色数が異なります。

アプリケーションツール「ライブスキャン」を標準装備

動画や静止画を簡単に保存できるアプリケーションソフト「ライブスキャン」※を標準装備。取り込んでいる映像を表示したり、残したいシーンを簡単に静止画保存したり、手軽な動画・静止画ハンドリングでパソコンの可能性をさらに広げます。X68030/X68000シリーズ用SX-WINDOW対応版とMacintoshシリーズ用QuickTime対応版の2種類を同梱しています。

※SX-WINDOW版はバージョン3.0以降（メモリー4MB以上）、QuickTime版はMacintosh漢字Talk7リリース7.1以上のシステムとQuickTime1.5以上（メモリー8MB以上）が必要です。

1,677万色対応の高速映像取り込み、動画・静止画の手軽なハンドリングが、新たなマルチメディアシーンを創造する。

SHARP INTELLIGENT VIDEO DIGITIZER CZ-6VS1　BUSY　POWER

■**SCSIインターフェイス採用**：パソコンの専用I/Oスロットを使わずに接続可能になり、汎用化を実現しました。またSCSI-2（FAST）インターフェイスの採用により、データ転送速度の高速化を図っています。X68030/X68000シリーズでは、SCSI-2（FAST）対応のハードディスクを接続することにより、パソコン本体を経由しないで、ハードディスクに直接、動画データをテンポラリデータとして記録することが可能です。パソコン本体のハードディスクへは、記録終了後に、テンポラリデータを変換し動画データとして保存できます。

※CZ-600C/601C/611C/602C/612C/652C/662C/603C/613C/653C/663Cに接続する場合は別売のSCSIインターフェイスボードCZ-6BS1ならびにSCSI変換ケーブルCZ-6CS1が必要です。※CZ-604C/623C/634C/644Cに接続する場合は、別売のSCSI変換ケーブルCZ-6CS1が必要です。
※Macintosh Power Bookシリーズに接続する場合は別売のSCSIケーブルなどが必要です。詳しくはMacintosh Power Bookシリーズの取扱説明書をご覧ください。

■**高機能MPUを搭載**：クロック周波数25MHzの32ビットMPU/MC68EC020を搭載、高速処理やパソコン本体の負担の軽減を実現します。

●MacはMacintoshの略称です。●Macintosh、Macintosh IIは、米国アップルコンピュータ社の登録商標です。●Power Bookは米国アップルコンピュータ社の商標です。●漢字Talk7はアップルコンピュータジャパン社の商標です。●QuickTimeは、米国アップルコンピュータ社の商標です。●価格には、消費税及び配送・設置・付帯工事費、使用済み商品の引き取り費等は含まれておりません。

ビデオ入力ユニット

CZ-6VS1

標準価格178,000円（税別）

CZ-6VS1の詳しい紹介や新作ソフト情報などX68シリーズの最新情報を満載。ちょっと役立つデータやプログラムなど、いろいろ楽しめるディスク情報誌をプレゼントします。
●EXE会員は、EXE会員番号と、90㎜（3.5型）/130㎜（5.25型）を明記の上、官製ハガキで左記「EXEディスク」係宛にお申し込みください。また、これからEXEクラブへ入会される方は、商品同梱のアンケートハガキに「EXEディスク希望」と明記の上、ご投函ください。（応募〆切 平成6年5月25日消印有効）
●EXEクラブに入っておられない方は、ソフトベンダー「TAKERU」での購入が可能です。（平成6年4月1日より2ヶ月間、予価200円）

C O N T

●特集

33 X68000と仲間たち

34 パワーアップアイテムあれこれ 周辺機器事情'94 中野修一

36 見果てぬ夢を目指して これが噂の040turbo 紀尾井誠

40 伝説のPOLYPHONは健在か? サブCPUシステムPOLYPHON-24 瀧 康史

42 高解像でSXは夢を見れるか? メガディスプレイへの道 瀧 康史

54 手軽さ・軽さ・大きさで見る 当代ハードディスク事情 伊澁見あきら

58 安全さを考えたインストール例 ちょっと大きめの常用HD 友永健明

59 選択から基本操作まで MOドライブの一般的使い方 中野修一

61 松下ドライブを使ったMOユニット 汎用MOユニットを接続する 紀尾井誠

62 CSM-120 Filo (コパル) 3.5インチ光磁気ディスクドライブの活用 丹 明彦

64 転ばぬ先の杖 満開式磁帯駆動装置壱號 友永健明

66 NSカルコンプ社Drawing Slate さらに進化したサイバー絵筆 川原由唯

68 ローテク工作実験室番外編 謎のPPIボードただいま見参 瀧 康史

74 パーソナルな「安心」 無停電電源装置BX3 紀尾井誠

●カラー紹介

14 特集 X68000と仲間たち

18 ショウレポート マイクロコンピュータショウ'94

●シリーズ全機種共通システム

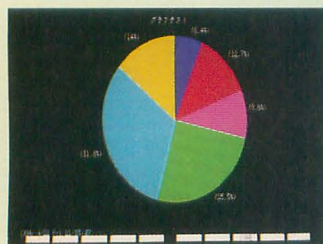
129 THE SENTINEL

130 YGCS ver.0.30 相沢栄樹

〈スタッフ〉
●編集長/前田 徹 ●副編集長/植木章夫 ●編集/山田純二 豊浦史子 高橋恒行 ●協力/有田隆也 中森 章 林 一樹 吉田幸一 華門真人 吉田賢司 朝倉祐二 大和 哲 村田敏幸 丹 明彦 三沢和彦 長沢淳博 司馬 護 清瀬栄介 石上達也 柴田 淳 瀧 康史 横内威至 進藤慶到 ●カメラ/杉山和美 ●イラスト/山田晴久 江口響子 高橋哲史 川原由唯 ●アートディレクター/島村勝頼 ●レイアウト/元木昌子 ADGREEN ●校正/グループごじら

特集 X68000と仲間たち

スーパーリアル麻雀PIV

あすか120% BURNING Fest.

F-Calc for x68k

マイクロコンピュータショウ'94

(で) のショートプロぱーてぃ

表紙絵：塚田　哲也

E　N　T　S

●THE SOFTOUCH

19　SOFTWARE INFORMATION 新作ソフトウェア/TOP10

GAME REVIEW
22　スーパーリアル麻雀PⅣ　清瀬栄介
24　あすか120% BURNING Fest.　瀧　康史
26　ジオグラフシール　八重垣那智
28　ストリートファイターIIダッシュ特別編　須藤芳政・日野麻也
30　大魔界村　横内威至

●読みもの

140　猫とコンピュータ　第91回 鬼婆白町から出られない　高沢恭子
142　第81回　知能機械概論―お茶目な計算機たち― V++という名のプロセッサアーキテクチャ　有田隆也

●続・こいのぼりPRO-68K

104　SX-BASIC公開デバッグ　第4回 開発キットとこいのぼりPRO-68K　石上達也
110　「PUSH BON!」のために作られた 外部関数の解説　朝倉祐二

●連載/紹介/講座/プログラム

16　響子 in CG わ～るど[第37回] アルジャーノンとチーズ　江口響子
76　ローテク工作実験室　第2回 パワースイッチ連動コンセントの製作　瀧　康史
80　ファイル共有の実験と実践(その8) 仮想ドライバの開発実験PART3.マルチドライバ化への挑戦　由井清人
84　新製品紹介 F-Calc for x68k　大和　哲
86　毎年恒例 第5回Oh!Xアンケート分析大会　高橋哲史
90　こちらシステムX探偵事務所 FILE-XIII フリッパーを作る　柴田　淳
98　(で)のショートプロぱーてぃ　その57 ショートプロってなんだっけ!?　古村　聡

113　Oh!X LIVE in '94
「スーパーストリートファイターII」より
キャミイのテーマ(X68000・Z-MUSIC+PCM8用)　平田恭敏
The End of Love(X68000・Z-MUSIC+ZPP.X用SC-55mkII対応)　鳥越英司

ハードコア3Dエクスタシー(第9回)
118　SIDE A　ドライビングモード詳説　丹　明彦
123　SIDE B　マップシステムの制作　横内威至

138　ANOTHER CG WORLD　江口響子

ペンギン情報コーナー……148
FILES Oh!X……150
質問箱……152
編集室から/DRIVE ON/ごめんなさいのコーナー/SHIFT BREAK/microOdyssey……158

1994 JUN. 6

UNIXはAT & T BELL LABORATORIESのOS名です。
Machはカーネギーメロン大学のOS名です。
CP/M, P-CPM, CP/Mupls, CP/M-86 CP/M-68K, CP/M-8000, DR-DOSはデジタルリサーチ
OS/2はIBM
MS-DOS, MS-OS/2, XENIX, MACRO80, MS C, WindowsはMICROSOFT
MSX-DOSはアスキー
OS-9, OS-9/68000, OS-9000, MW CはMICROWARE
UCSD p-systemはカリフォルニア大学理事会
TURBO PASCAL, TURBO C, SIDEKICKはBORLAND INTERNATIONAL
LSI CはLSI JAPAN
HuBASICはハドソンソフト
の商標です。その他，プログラム名，CPU名は一般に各メーカーの登録商標です。本文中では"TM", "R"マークは明記していません。
本誌に掲載されたプログラムの著作権はプログラム作成者に保留されています。著作権上，PDSと明記されたもの以外，個人で使用するほかの無断複製は禁じられています。

■広告目次

アルバトロス……167(下)
計測技研……168
シャープ……表2・表4・1・4-7
九十九電機……162-163
ネオコンピュータシステム…167(上)
P & A……164-165
ヒューマンクリエイティブスクール……8
ブラザー工業……表3
BLUE SKY……166
満開製作所……160

フィールドが、膨らむ。

先が、ますます面白くなる。

●

未来への確かなビジョンをベースに
発展性のあるプラットホームとしてのウィンドウ環境を提供する
国産オリジナルウィンドウシステムSX-WINDOW。

●

GUI環境や操作環境、高速化へのゆるぎない探求、
マルチメディアの統合的なハンドリング。

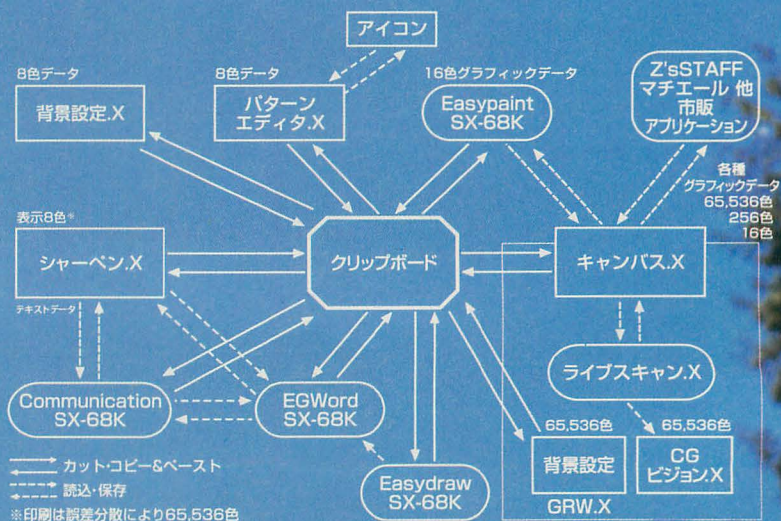
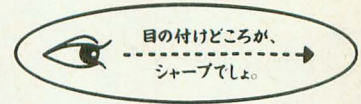
●

いま、より多彩なフィールドへ
そのインテリジェンスが展開を始める。

●

次のステージが見えてくる。

SX-WINDOW ver.3.1のデータ利用環境

目の付けどころが、シャープでしょ。

今も、先も楽しめる。
いつも新展開の予感、SX-WINDOWのニューバージョン。

❶ マルチフォントエディタ編集例。文字ごとに文字種・文字の大きさの指定、装飾が可能。

❷ 用途に合わせてカスタマイズできる便利な「エディタ」モードの例。

❸ プリンタ印字例（カラー印刷は誤差分散により65,536色対応。）

❹ 「パターンエディタ」で作成したデータを背景に設定可能。

❺ 日本語フロントプロセッサASK68K Ver.3.0の辞書メンテナンスがウィンドウ上で可能。

❻ アイコンデータや背景データを作成する「パターンエディタ」。

❼ オリジナルに作成したアイコンパターンの例。

❽ 512×512ドットの範囲内で65,536色の表示が可能。

❾ さまざまなグラフィックフォーマットに対応。

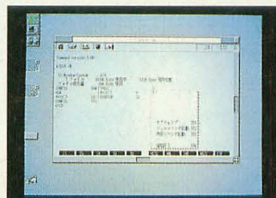
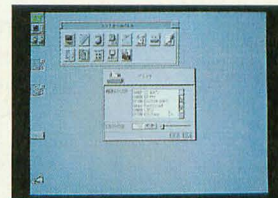
❿ 任意のサイズに縮小・拡大表示が可能。

⓫ 異なる画像フォーマットへのコンバートが可能。

⓬ 別売のビデオ入力ユニットCZ-6VS1(標準価格178,000円)を介して映像ハンドリングが可能。

■実画面:1,024×1,024ドット,表示画面:768×512ドット

●画面は広告用に作成した、機能を説明するためのイメージ画面です。また、各種アイコンなどは、SX-WINDOW ver.3.1がもつ機能を使って作成したもので、標準装備のものとは異なるものもあります。
●本広告中のエディタで表示している文字のフォントはツァイト社の、「書体倶楽部」のフォントを使用しています。

●インライン入力のサポート：ASK68K Ver.3.0を利用したインライン入力をSX-WINDOWで実行可能。またシャーペン.Xをワープロとして利用できるよう、さまざまな機能が付加されています。

●コンソールをサポート：Human68kやX-BASICのコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できます。
（グラフィックを利用したものなど、SX-WINDOWと処理が重複するものは実行できません。）

●多彩なプリンタに対応：さまざまなSX-WINDOWアプリケーションで利用できるページプリンタドライバを標準装備。ESC/Page、LIPS III、PostScriptに対応したプリンタが利用できます。

SX-WINDOW ver3.1

CZ-296SS(130㎜FD)/CZ-296SSC(90㎜FD) 標準価格22,800円(税別)

●Z'sSTAFF、書体倶楽部はツァイト社の商標です。●ESC/Pageはセイコーエプソン株式会社の登録商標です。●PostScriptはアドビシステムズ社の登録商標です。●EGWordは株式会社エルゴソフトの登録商標です。

■お問い合わせは… シャープ株式会社 電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

資料請求券 ペリフェラル Oh!X 6係

◎パーソナルDTPをX68で

XDTP SX-68K

CZ-291BWD 6月発売予定

NEW

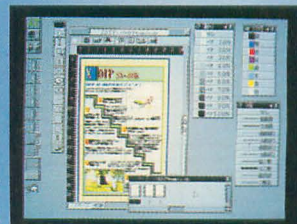
縦書きをはじめとした多彩なレイアウト機能で
パーソナルなデスクトップパブリッシングを実現するソフトです。
やさしい操作、豊富な編集機能、グラフィックウィンドウ対応、SX-WINDOWをすでに
ご利用になっている方なら、基本操作を新たに覚えることなく手軽にレイアウトが作成できます。

■豊富なテキスト編集機能：フォント種類、サイズ、文字種の変更はもちろん、上線、下線、網掛け、文字間隔の指定が文字ごとに設定可能。禁則、行間隔、タブ、インデント、マージンもパラグラフ(リターンコードまでの文字列)ごとに設定できます。また各テキストフレームごとに、フレーム形状、リンク状態(テキストの流し込み)、縦書き/横書き、回り込みの設定が可能。検索/置換も単純な文字列だけでなく、スタイル別に行うことができます。

■グラフィックウィンドウに対応：GRW.Xにも対応していますので、いろいろな形状でレイアウトしたグラフィックフレームのデータを65,536色の画像で確認しながらレイアウトできます。

■さまざまな画像フォーマットに対応：ビデオマネージャが対応している静止画フォーマットの他に、「PrintShop PRO-68K」、「CANVAS PRO-68K」、「GScriptファイル」の読み込みに対応しています。

●グラフィックフレーム、テキストフレームとは別に直線、矩形、楕円、多角形が作成できる独立した罫線機能●第1水準を収めた明朝体、ゴシック体のベジェーフォントファイルを標準装備●ページの移動や作成/削除がスピーディに行える独立したページウィンドウをサポート●ページプリンタドライバ(ESC/Page、LIPSIII)を付属、高解像度の美しい印字が可能。またSX-WINDOWが対応しているプリンタも使用可能。

※5MB以上の空きのあるハードディスクが必要です。 (4MB、Ver.3.0)

◎グラフィック感覚の楽譜入力をサポート

MUSIC SX-68K

CZ-274MWD 標準価格38,000円(税別)

NEW

MIDI、FM、ADPCMに対応した楽譜ワープロ&作曲演奏ソフトです。
自由なレイアウトでグラフィックを描くように楽譜入力、
全パートの同時入力や編集、自動伴奏機能、応用範囲を広げるデータ互換性。
多彩なプリンタ対応で美しい印刷も可能です。

■MIDI、FM、ADPCM対応：MIDI、FM、ADPCMを同時に発音できます。全ての音源を利用した場合、最大発音数は25まで設定可能です。

■全パートの同時入力：ピアノ譜、メロディ譜などの組み合わせで最大16パートまで編集可能。特定パートごとではなく全パートを画面に表示して編集できますので、直接画面上で曲の構成を考えながら作編曲できます。

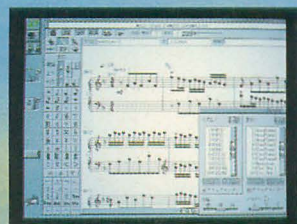
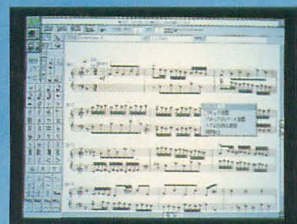
■コード&リズムによる自動伴奏機能：メロディ上にコードネームとリズムパターンを入力するだけで、自動的に伴奏をつけることができます。

■優れたデータ互換性：「MUSIC PRO-68K」、「MUSIC PRO-68K[MIDI]」のデータファイルが利用できる他、OPM、MML、ZMSファイル形式でデータ出力が可能です。

■多彩なプリンタ対応：ページプリンタドライバ(ESC/Page、LIPSIII)を付属、高解像度の美しい印刷が可能です。
またSX-WINDOWが対応しているプリンタも利用できます。

(4MB、Ver.3.0)

For
X68030/
X68000series
APPLICATION
SOFTWARE

X68030
32bit PERSONAL WORKSTATION

その先のシーンへ。

●さらに実用的なウィンドウシステムへの進化

SX-WINDOW ver3.1 システムキット

CZ-296SS(130mmFD)/CZ-296SSC(90mmFD)　標準価格22,800円(税別)　NEW

ASK68K Ver3.0を利用したインライン入力のサポート、Human68k/BASICコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できるコンソールのサポートをはじめ、シャーペン.Xをワープロとして利用できるよう機能アップ。また、さまざまなSX-WINDOWアプリケーションで利用できるページプリンタドライバを標準装備。ドローデータ(FSX)/フォントデータ(IFM)処理の高速化も実現しています。

※コンソールでは、SX-WINDOWと処理が重複するものは実行できません。　4MB

※既にSX-WINDOWをお持ちの方には有償バージョンアップサービスを行います。

●定評のGUI対応ウィンドウワープロ

EGWord SX-68K

CZ-271BWD　標準価格59,800円(税別)　NEW

ウィンドウワープロとして評価の高いEGWordのSX-WINDOW対応版。キャラクタベースのワープロを超えたグラフィカルユーザーインターフェイス(GUI)による手軽なDTPソフトとしても優れた表現力を発揮します。定評ある日本語入力方式(EGConvert)によるインライン入力、さまざまなグラフィックデータ(GScript)やテキストデータの貼り込み、また文書互換を実現するEDF(Extended Document Format)形式をサポートしています。　4MB、ver.2.0

●SX-WINDOW開発支援ツール

SX-WINDOW 開発キット Workroom SX-68K

CZ-288LWD　標準価格39,800円(税別)　NEW

SX-WINDOW用のソフト開発に必要なツールやサンプルプログラムを装備。プログラムの編集、リソースの作成、コンパイル、デバッグといった一連の作業をSX-WINDOW上で効率よく実行できます。初めてSX-WINDOW用のプログラムに挑戦する人にも、簡単に基本機能の理解が深まる33種(基礎編23種、応用編4種、実用編6種)のサンプルプログラム付き。

※ご使用に当ってはC compiler PRO-68K ver.2.1が必要です。　4MB、ver.2.0

●SX-WINDOW開発キットのサポートツール

開発キット用ツール集

CZ-289TWD　標準価格12,800円(税別)　NEW

SX-WINDOW開発キットをさらに使いやすくするためのツールです。SXコールの簡易リファレンスを簡単に検索するインサイドSX、イベントの発生を常時監視・確認するイベントハンドラ、リアルタイムにメモリブロックの利用状況を表示するヒープビューアなど11種のツールが用意されています。　2MB、ver.2.0

●SX-WINDOW対応ドローイングツール

Easydraw SX-68K

CZ-264GWD　標準価格19,800円(税別)

イラスト、フローチャート、地図、見取り図など各種グラフィックが製図感覚で作成できます。作成したデータは他のSX-WINDOW対応アプリケーションでも利用でき、企画書などの作成をサポート。ページプリンタドライバも標準装備。　4MB、ver.3.0

●ウィンドウ対応グラフィックツール

Easypaint SX-68K

CZ-263GWD　標準価格12,800円(税別)

マウスによる簡単操作、65,536色中16色の多彩な表現、クリエイティブマインドに応えるウィンドウ対応ペイントツールです。同時に複数のウィンドウを開いて編集でき、各ウィンドウ間でのデータ交換もできます。　2MB、ver.1.1

●SX-WINDOWを楽しく使うためのアクセサリ集

SX-WINDOW デスクアクセサリ集

CZ-290TWD　標準価格14,800円(税別)

SX-WINDOWをさらに便利に楽しく使うためのデスクアクセサリ集です。スクリーンセーバ、スクラップブック、スケジューラ、アドレス帳、電子手帳通信ツール、パズルなど、12種の豊富なアクセサリが収められています。　2MB、ver.3.0

●マルチタスク機能をはじめ通信環境がさらに充実

Communication SX-68K

CZ-272CWD　標準価格19,800円(税別)

通信環境をさらに高めたウィンドウ対応の通信ソフトです。マルチタスク機能により他のアプリケーションを実行中でも簡単に通信が可能。自動ログイン機能やプログラム機能、など豊富な機能をサポートしています。　2MB、ver.1.1

●FM音源サウンドエディタ

SOUND SX-68K

CZ-275MWD　標準価格15,800円(税別)

他のミュージックソフトで演奏中の音色を、簡単に作成、変更できるマルチタスク機能、またエディット、イメージ、ウェーブの3つの編集/確認モードを装備。作成中の音色も50曲の自動演奏でリアルタイムに確認、編集できます。　2MB、ver.1.1

●SX-WINDOW対応になってさらにパワーアップ

倉庫番リベンジ SX-68K ユーザー逆襲編

CZ-293A(130mmFD)/CZ-293AWC(90mmFD)　各標準価格6,800円(税別)

倉庫番10年にわたるユーザーの投稿など、新作306面が目白押し。まさに倉庫番の最強版がSX-WINDOW上で楽しめます。AI機能やエディット機能、キャラクタ変更機能も装備。半年で解けたらあなたは天才?です。　2MB、ver.1.1

PRO-68K シリーズ

●X68030/X68000対応

C COMPILER PRO-68K ver.2.1 NEW KIT

CZ-295LSD　標準価格44,800円(税別)

C compiler PRO-68KのX68030/X68000対応版。MPU68030、MC68882の命令セットに対応したアセンブラ、デバッガ、ソースコードデバッガを付属。またHuman68k ver.3.0、ASK68K ver.3.0にも対応。新たにGPIBライブラリ、MC68882対応フローートライブラリを付属しています。

※メインメモリ2MB以上が必要です。

※C compiler PRO-68K/ver.2.0/ver.2.1をお持ちの方には有償グレードアップサービスを行います。

※(2MB,ver.1.1)の表示は、メインメモリ2MB以上、SX-WINDOW ver.1.1以上が必要であることを示します。

※発売予定のソフトの画面は実物とは異なる場合があります。

●EGWord、EGConvertは株式会社エルゴソフトの登録商標です。●ESC/Pageはセイコーエプソン株式会社の登録商標です。

●お問い合わせは…シャープ㈱電子機器事業本部(液映)システム機器推進プロジェクトチーム　〒162　東京都新宿区市谷八幡町8番地　☎(03)3260-1161(大代表)へ　シャープ株式会社

資料請求券

HUCEC'94

求む、蝶 超発想!!

募集部門

1.プログラム部門
2.CG映像部門
3.ミュージック部門
4.企画・アイディア部門

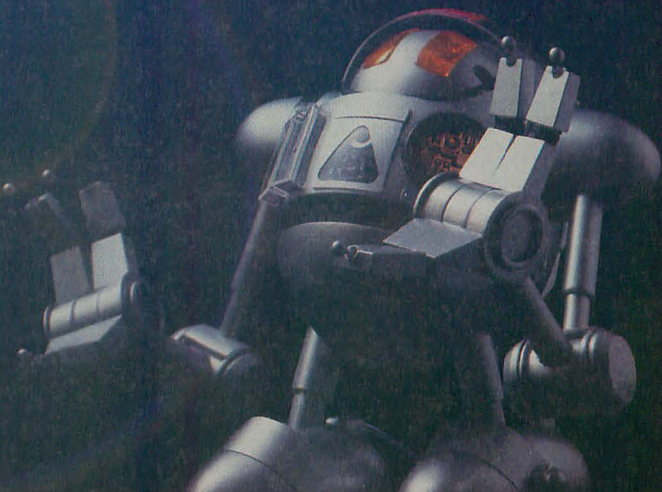
ヒューマン コンピュータ エンタテインメント・コンテスト '94

HUMAN COMPUTER ENTERTAINMENT CONTEST '94

現在コンピュータを利用した創作環境は多様化しつつあり、かつ、その環境は個人レベルでも利用可能となってきました。ヒューマンクリエイティブスクールでは、将来のコンピュータ・エンターテイナーの登竜門となるべく場を提供したいと思います。「才能のある人達にスポットライトを当て、新しいエンタテインメントを創作する道を拓くこと」これが本コンテストの開催理念です。

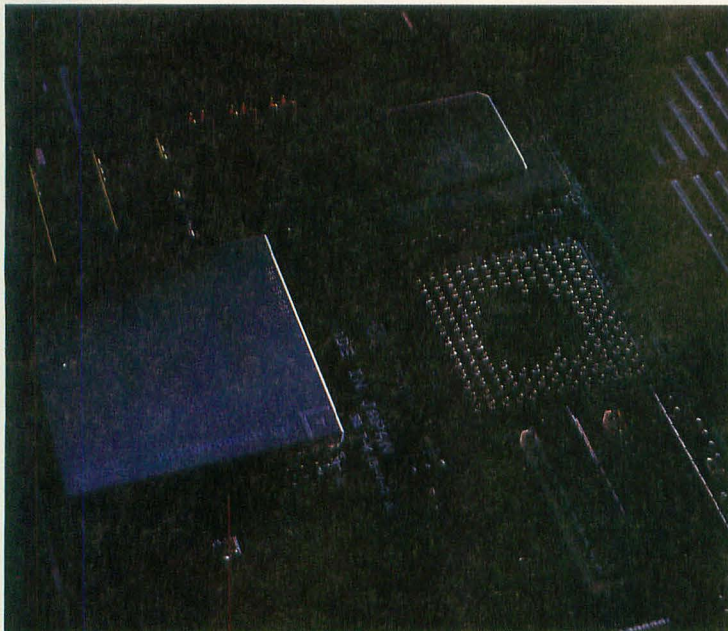
- 応募締切／平成6年 9月30日（当日消印有効）
- 応募作品／未発表のもので自作のもの。個人・団体応募可。
- 著作権／応募作品の著作権は、応募者に帰属致します。
- 入　賞／グランプリ賞金100万円
  ※その他、部門賞、各社協賛企業賞あり。
- 発　表／平成6年度、ヒューマンクリエイティブスクール学園祭（平成6年11月開催）にて、審査結果発表及び授賞式。

主催／ヒューマンクリエイティブスクール
共催／株式会社日本パーソナルコンピュータソフトウェア技術研究所
後援／社団法人日本パーソナルコンピュータソフトウェア協会
財団法人ソフトウェア工学研究財団
協賛／エプソン販売株式会社／キヤノン販売株式会社／シャープ株式会社
株式会社ソニー・コンピュータエンタテインメント／パイオニアLDC株式会社

※応募要項を下記宛にご請求ください。

お問い合せ先／ヒューマンクリエイティブスクール内　HUCEC'94運営事務局　Tel. 0422-23-1111　〒180 東京都武蔵野市吉祥寺北町1-2-12

X68040turbo

A Story of Making "After X68030"

BEEPs◉著　●B5変形判●定価2,400円

新刊

X68030が68040マシンに変わる！

ウワサの040turbo、ついに登場。

040turboとは何なのか？

何ができ、どのくらいのパワーを秘めているのか？

製作者自身が明かす

040turbo誕生のストーリー、

ここに登場。

(編)コーナー

心からX68kを愛する著者が挑んだ、血湧き肉躍る68040ボード開発ストーリー。読物としての面白さとともに、MPU68030、68040の生きた解説書としても十分役に立ちます。本書を読めば、X68kユーザであったことを心から誇らしく思えるはずです。

X68030 Inside/Out

新刊

桒野雅彦◉著
●B5変形判・200ページ
●定価3,000円

X68030のハードウェアの特徴を、X68000と比較しながら、
本体内部と外部にわたって解析した本。さらに、『Inside X68000』執筆後、
筆者が見つけた機能についても補足してあります。
巻末には、X68030では利用できませんが、
要望の強いMMU機能についても解説しました。X68030回路図付き。

X68k Programming Series #3

X680x0 TEX

続刊

吉野智興・川本琢二・山崎岳志・実森仁志◉著
●B5変形判・2冊組・ビニール箱入り●5"FD8枚組

待望久しいX680x0版TEXの解説書。
はじめてTEXを使いたい人には簡単インストーラを付けてあるほか、
すでにTEXを使っている人には自分の環境にあわせた
カスタマイズのしかたを説明してあります。また、リファレンス編では、
TEX, fontman, preview, print, makefont, METAFONTなどの
環境変数・オプション等の説明をまとめておきました。

(編)コーナー

遅れています。lips 3dviなどのTEX周辺プログラムをまとめたディスクを付けることにしましたので、ディスクの枚数が1枚増えてしまいました。もう少しお待ちください。

SOFT BANK ソフトバンク株式会社 出版事業部　〒103 東京都中央区日本橋浜町3-42-3 電話03(5642)8101

GCCによるX680x0
ゲームプログラミング

吉野智興　著

定価3,600円

5"2HDフロッピィー×2枚

(GCC、GDB、HAS、HLK、LIBC収録)

本書は、X68000／X68030ユーザを対象に、コンピュータの基礎知識から、C言語の入門、ゲームプログラムの作成までを、分かりやすく解説した実践的なCプログラミングの入門書である。「付録ディスク」には、本書の全ソースプログラムと、それをコンパイル／リンクするための実行環境(GCC、LIBC、etc)を収録している。
初めてCを学ぶ初心者から、ゲームプログラミングに関心を持つ、中上級者まで、すべてのX68000／X68030ユーザに最適の1冊である。

目次より

❶……………………ゲームプログラミング入門
❷……………………………C言語入門
❸…………………ゲームプログラミング基礎知識
❹……………………………C実践ゲーム製作

SOFT BANK

ソフトバンク株式会社／出版事業部

Z-MUSIC システム Ver.2.0

ついにMUSICシステムの正式バージョンアップ版が登場します。
X68000の音源ドライバとしてさらに使いやすく高機能なものになりました。

ver.1.0/1.1からのバージョンアップ内容

PCM8対応AD PCM同時発音8声音量可変
モジュレーション用波形メモリ搭載
PCMバンクに対応
ステップエディット系コマンド追加
X68030完全対応ユニバーサルバージョン
RS-232C対応版収録
POLYPHON対応版収録
再生専用機能縮小版収録
Cコンパイラ用ライブラリ完成
AD PCM加工機能強化
さらにクオリティを高めたAD PCMデータ
もちろん、全ソースプログラム付属&ライセンスフリー

SOFT BANK ソフトバンク株式会社／出版事業部
〒103 東京都中央区日本橋浜町3-42-3 TEL03-5642-8100

スーパーリアル麻雀 PIV

アーケード版 ＋プラス スーパーファミコン版

原画&設定資料集

●(株)セタ/監修
●田中 良 /画

A3描き下ろしポスターつき

KAORI

YOU

ファン待望!! 初の公式ブック登場

アニメを見るかのような滑らかなグラフィックでそれまでの単なる麻雀ゲームの常識を打ち破った「スーパーリアル麻雀」シリーズ。PIIではじめてかわいい美少女キャラが登場して以来、現時点での最新作PIVまで、常にトップの人気を集めてきた。本書はそのPIVの魅力を全部公開するはじめての公式設定資料集だ。アニメーターとして活躍中で、シリーズすべてのキャラクターデザインおよび原画を担当してきた田中良氏の血と汗と努力が結集した原画の数々と描き下ろしイラストをとくとご覧ください。

定価2,000円

MANA

©1993 SETA CO.,LTD.

SOFT BANK ■定価は税込 ■お求め・ご予約はお近くの書店で
ソフトバンク株式会社／出版事業部 販売局：TEL.03-5642-8101

スーパーガイドシリーズ

ファイアーエムブレム〜紋章の謎 スーパーガイド

Theスーパーファミコン編集部特別編集

RPG要素をくわえたファンタジーシミュレーションの名作が、スーパーファミコンでグレードアップして登場。第1部と2部の全44マップを徹底解説し、戦略に必要なアイテム・ユニットデータを完璧にフォロー。また、ゲームに登場するキャラクター90名の事典つき。これ一冊で、ゲームの世界を完全把握できるぞ。

A5判・定価780円

スーパーロボット大戦EX スーパーガイド

スタジオエル　編著

スーパーファミコンの人気シミュレーションゲーム「スーパーロボット大戦ＥＸ」の完全攻略本。ゲームに登場する全マップを紹介し、シナリオ別フローチャート、全ユニットの完全データ表など、読めば得する情報を満載しました。また、本書の特徴である〝タイプ別攻略法〟によって、読者の皆さんは自分流のゲームの楽しみ方を発見できるでしょう。

A5判・定価550円

【シリーズ既刊】……………… 好評発売中！

ストリートファイターIIターボ スーパーガイド

Theスーパーファミコン編集部特別編集

大ヒット中の「ストリートファイターIIターボ」の完全攻略本。全12人の登場キャラクター別にそれぞれの対戦攻略を徹底ガイド。新必殺技など基本技以外の決め技を大紹介。

A5判・定価650円

スーパーマリオコレクション スーパーガイド

Theスーパーファミコン編集部特別編集

人気ソフト「スーパーマリオコレクション」の完全攻略ガイドブック。「スーパーマリオブラザーズ」および2・3、そして「スーパーマリオUSA」という4本の大人気ソフトをマップつきで徹底解説。

A5判・定価890円

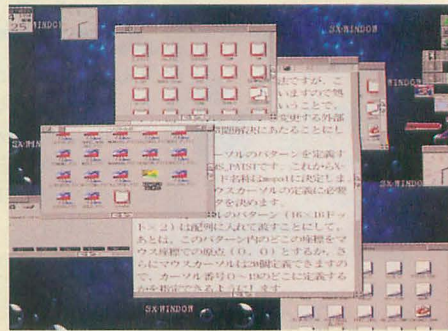
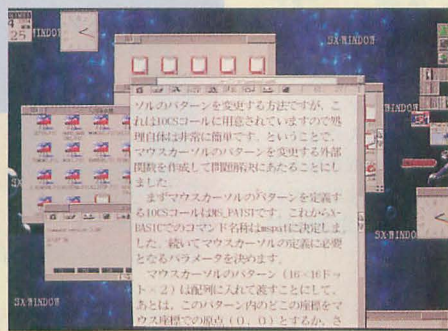
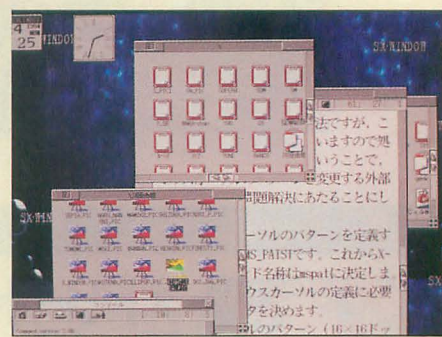
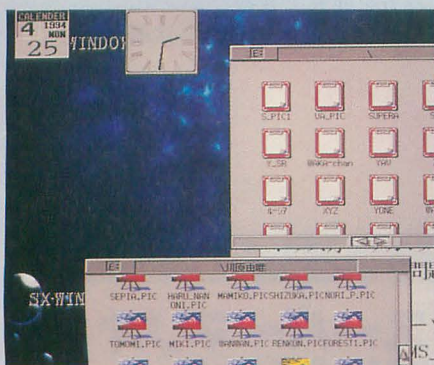
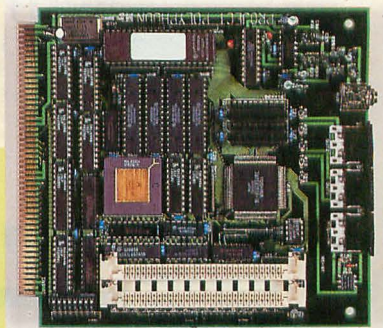
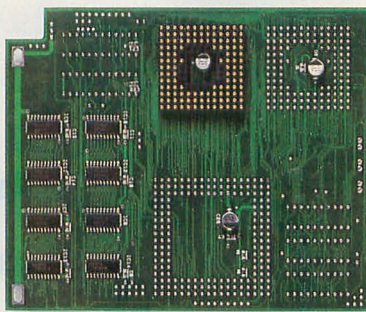
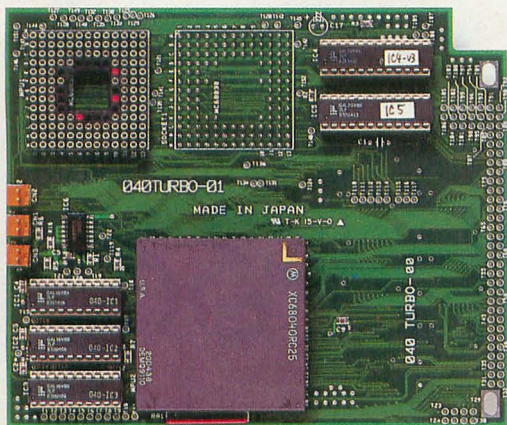
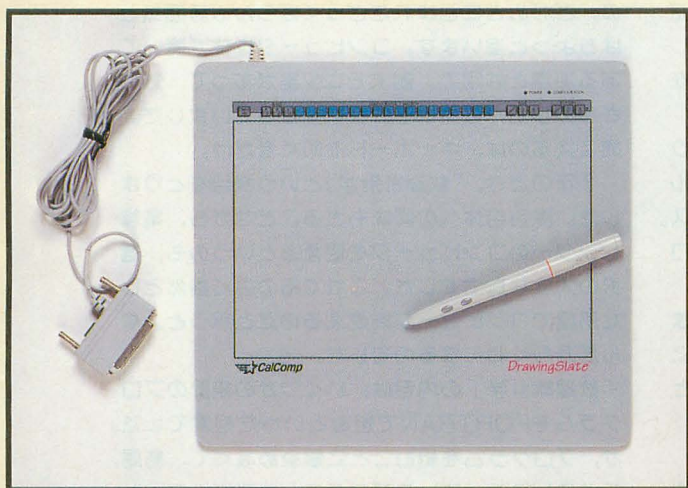
◆定価は税込みです　◆お近くの書店でお求めください

ソフトバンク株式会社/出版事業部 SOFT BANK
販売局 TEL.03-5642-8101

[特集]

X68000と仲間たち

周辺機器がかんばって，ツールがかんばって，ユーザーがかんばると
見えてくる世界がある。あんなこともこんなこともできる
そんな三位一体の世界を垣間見てみよう

これぞ「満願成就的映像美」といえる。ついにSX-WINDOWで1024×1024ドットモードが達成された。ドット比は1：1ではなくなっているが，背景のグラフィックは横長ドットで違和感はない。縦横各2倍だから当然といえば当然か。推奨CRTはCR-7000だ。ちなみに撮影に使用した機材ではクロックオシレータを69.5MHz→80MHzに変更してある。無改造の場合は横が広がってはみ出してしまう。
下は画面モードを変えたときのSX-WINDOWデスクトップの様子。順に512×512ドット，768×512ドット，880×660ドット，1024×768ドット，1024×800ドット，1024×1024ドットモードだ。
通常のディスプレイだと1024×768ドットあたりが限界となる。

左が040turboの基板だ。68030用ソケットの大きさと比べてみると68040チップの大きさがわかると思う。実際に使用する際にはヒートシンク＋強制冷却措置が必須と考えたほうがよいだろう。中央の写真は040turbo裏面。両面に部品が実装されていることがわかる。右は高密度実装のPOLYPHONボードだ。

川原画伯タブレットを使う第2弾。色の濃淡や線の強弱を筆圧でコントロールできる。もちろんそのまんま手描き感覚だから，筆のタッチや鉛筆の風あいなども思いどおりにつけられる（かもしれない）。

下はMATIERのオートペイント機能の実行例。写真などをもとにすれば，ブラシを当てるだけで油絵風，水彩画風その他の表現が手軽にできる。

[第37回]
アルジャーノンとチーズ

江口 響子

響子inCGわ～るど

アルジャーノンは，ダニエル・キイス原作の小説「アルジャーノンに花束を」に出てくる実験用のハツカネズミの名前です。脳外科手術を受けて賢くなった彼は，知能テストとして箱の中の迷路を，走り回ります。ゴールに置いてある，おいしいチーズを目指して。チーズは彼の動機。迷路を解くのは，チーズがほしいからなのです。

さて，いま私の手元には，3つのマウスがあります。X68000用の普通のマウスとトラックボールにもなるマウス，それにMacintosh用のマウス。この3匹は，ゲーム，CG制作，表計算にワープロと，なくてはならぬ存在です。

コンピュータにマウスがついているのは，いまではあたりまえのこと。けれど，私が大学2年になって初めてコンピュータというものに触れたとき，そこにマウスはありませんでした。

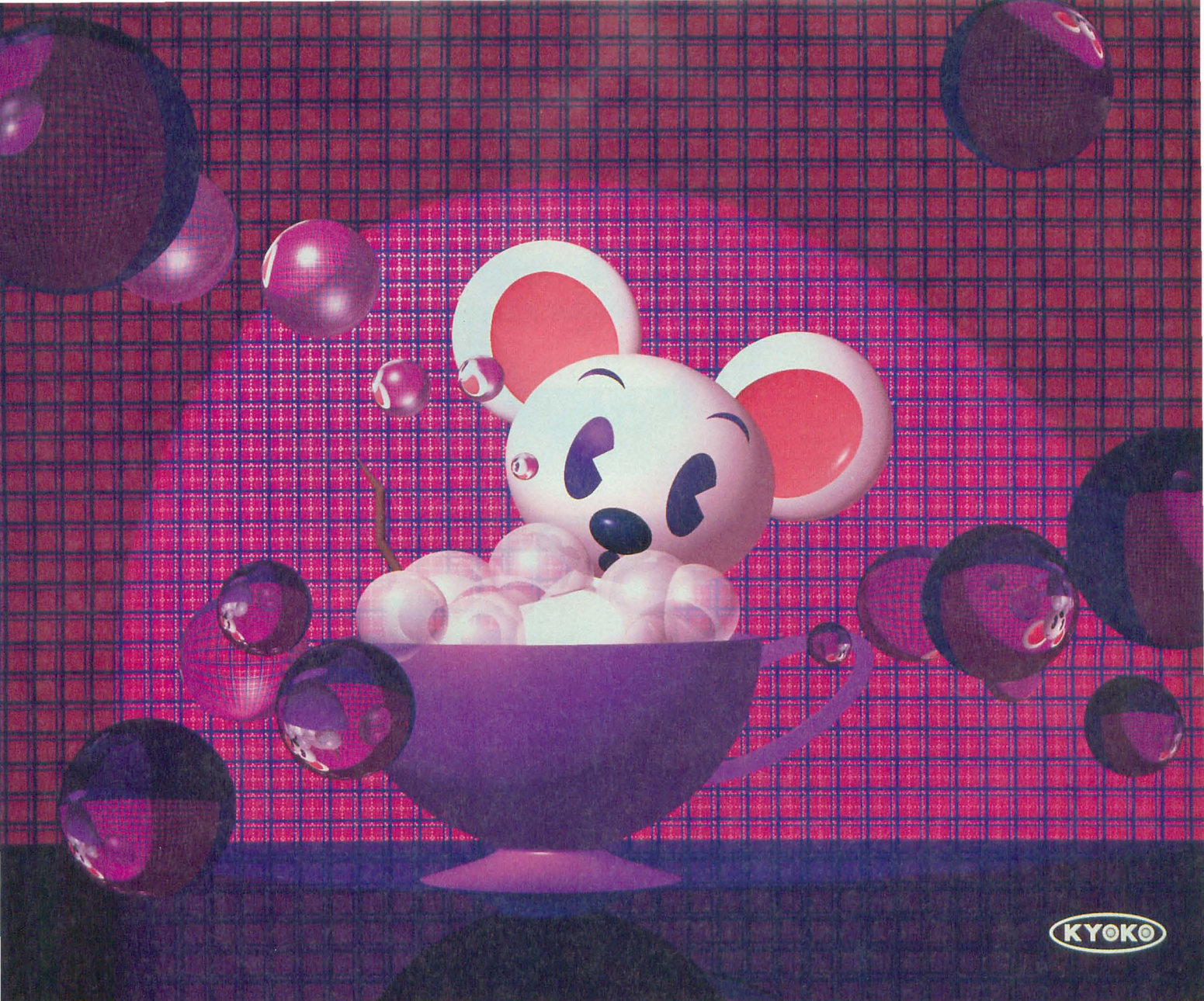
大学内に電子計算機センターというのがあり，日立製の大型コンピュータHITACが動いていました。数十台の端末からの命令を時間順に受けつけて，1台のコンピュータで処理するタイムシェアリングシステムが当時の売り物でした。センターは，病院のように白く清潔で，なんだか空気までもが澄んでいるようでした。ほかの建物や教室の，ざわめきに包まれたざっくばらんな雰囲気とはちょっと違います。コンピュータ本体が置いてある部屋とは別に，端末モニタ室があって，たくさんの学生が整然とモニタに向かっていました。聞こえるのは，キーボードを叩く音だけ。

3年のとき，「数理統計学」という課目をとりました。課目自体への興味もさることながら，電算センターのコンピュータを使えるというのも，選択の大きな理由でした。これであの白く整然とした部屋でコンピュータを使えるのだと思うと，なんだかわくわくするのでした。

「数理統計学」の内容は，いくつかの統計のプログラムをFORTRANで組むといった程度でした。が，プログラムを組むことに馴染めなくて，結局，テキストに載っているプログラムを適当にアレンジするだけの情けない結果に終わってしまいました。

当時，CGは巷の話題になっていましたが，やはりプログラムが必要というので，とうてい自分には無理とあきらめていました。コンピュータと対話する方法が，キーボード入力のプログラムだけというのは，初心者にとってなかなかつらいものです。

コンピュータは私には向かないと思いはじめたころ，マウスで絵を描くソフトがパソコンにあるのを知りました。友人の持っていたMacintosh Plus。ああ，こんなにコンピュータって簡単に使えるのかというのが実感でした。それがきっかけで，パソコンとのつきあいが始まり，PC-8801，MacintoshSE/30，PC-9801シリーズを経て，いまのX68000シリーズ，Macintosh ColorClassicと続いています。

パソコンと向き合っている時間が長くなると，しだいにコンピュータそのものに興味が出てきます。最初は，マウスで絵を描くのがせいいっぱいだったのが，OSを使って環境を整えることが面

白くなります。そして，必要に応じて，簡易言語や簡単なプログラムを使うようになり，あきらめていた３次元CGも，パソコンで使えるソフトが出たおかげで，こうしてみなさんに見ていただけるようになっています。

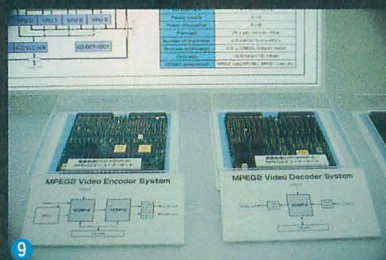
アルジャーノンは，チーズを目指して迷路を解きました。先になにがあるかわかっていたからです。コンピュータとのつきあいは，まるで迷路のようです。あっちにぶつかり，こっちで迷いながら，しかもゴールになにがあるのか，そもそもゴールがあるのかもわかりません。でも，この迷路にはイベントがいろいろ隠されていて，中にいるのが楽しいのです。

コンピュータに親しむきっかけを与えてくれたマウスたち。いま，そばにいる３匹。Macintoshは，去年の暮れに買ったので新しいのですが，X68000のほうは５年も使っているので，ずいぶんと手垢で汚れています。本当は，すみずみまでお湯で，きれいに洗ってあげたいところです。

注：中編「アルジャーノンに花束を」は，「心の鏡」ダニエル・キイス傑作集（早川書房）に収録されています。この傑作集には，人工知能や臓器移植を題材とした作品なども あって充実しています。また，中編を細かく書き込んだ長編は，「アルジャーノンに花束を」（早川書房）として出ています。こちらのほうはすでに読まれた方も多いと思います。まだの方は，ぜひご一読を。

今回のCGデータ

1920×1536ピクセル
1670万色フルカラーを４×５ポジで出力
使用ソフトはサイクロン。総物体数78　スポット光１
マッピングデータは，Z's STAFF PRO-68Kで作成

マイクロコンピュータショウ'94

①ここが会場となった東京流通センター　②シャープではEXPAND PADの分解された姿も　③なぜか元気な日立ブースではSH7600のミニコンファレンスが行われていた　④高速なシンクロナスDRAMを使ったグラフィックデモ　⑤これが最近話題のPOWER PCチップだ　⑥なんとこれはAlphaAXPチップ搭載のノートパソコンだ　⑦東芝のR4600　⑧手書き感覚のB-TRON用タッチパネル　⑨MPEG2に対応したパナソニックのDSPチップとMPEG2エンコーダ/デコーダ　⑩そのデコーダを使用したMPEG2による動画像の例。さすがに綺麗だ

4月19日から22日の4日間，東京流通センターでマイクロコンピュータショウ'94が開催された。テーマは「マルチメディア時代に向けて」となっており，マルチメディア機器のコアとなる技術を中心に出展されていた。

スペースは例年の40%程度，出展者数はさらに大幅に減っており，昨年と比べてもかなり規模が縮小され，不景気を反映してか，少しさみしい状況での開催となった。中小のシステムハウスクラスの出展はほとんどなくなっていた。

シャープではマルチメディア時代の基礎技術を展示し，マルチメディアに対する意欲を見せた。同社では今後，携帯型情報端末を中心としたマルチメディア戦略を展開していく模様。

関連製品としてNEWTONにふたをつけたEXPAND PADやペンコムZAURUS，液晶ビューカムなどの製品が展示されていた。

基礎技術関係ではROMとRAMを混在させたニューメモリチップやARM社のRISCチップ，カラー液晶などを展示。

他社ではRISCチップ関連が目立った。

特に元気だったのは，いわゆる「SH2」チップSH7600シリーズを発表した日立。小型で高速，低消費電力，高機能をアピールしていた。SH7600はRISCチップだが，周辺回路をチップ上に多く取り込んでいるのが特徴的。また，DSPのような積和演算命令を備えるなど，小回りのきくCPUだ。シンクロナスDRAMとの組み合わせでコストパフォーマンスの高いシステムができることなどを強調。そのパフォーマンスはセガSATURNで明らかにされることだろう。

ハイエンド分野ではDECのAlpha AXPが目立っていた。DECchip21064Aは128ビットの外部バス幅を持つ64ビットのRISCチップでスーパースカラーにより最大2命令同時実行をこなす。最大275MHzで駆動され550MIPSを叩き出すという。

そのほか，低価格/高性能をウリにした東芝のR4600はR4000のキャッシュ容量を増やしたものだ。これはSGIの新しいIndyに搭載されたチップである。

IBMはもちろんPOWER PCだ。展示されていたのは601と603のみで，若干もの足りないがPOWER Macintoshですでに身近なところまできているためアピール性は強い。

そのほか，お馴染みのTRON協会は，IBM互換機用のB-TRONやタッチパネルをユーザーインタフェイスにしたGUIなどを展示していた。

ソフトウェアの切り換えによりMPEG2デコードとエンコードを1チップでこなすパナソニックのDSPチップなども今後の基幹技術となるだろう。

SOFTWARE INFORMATION

陽光眩しい5月をすぎて，6月はちょっとうっとうしい季節。でもX68000ユーザーのキミなら大丈夫。ゲームの新作はあるし，シャープからも各種ソフトが続々と登場予定。Oh!Xも強い味方さっ！（なんてね）

Mr.Do!/Mr.Do! VS UNICORNS

安定した人気を誇るビデオゲームアンソロジーシリーズ。はやくも第10作目のタイトルが決定した。今回は2つのアクションゲームがパッケージングされている。

ひとつ目の「Mr.Do！」は1982年のユニバーサルの作品。主人公のMr.Doを操作して，敵から逃げつつ穴を掘って画面のあちこちに散らばっているサクランボを集めるゲーム。リンゴを落としたり，石を投げたりして敵をやっつけるのだ。

もうひとつの「Mr.Do！ VS UNICORNS」は同社の1983年の作品。前作と同じく主人公はMr.Do。しかし，ゲーム内容は違っている。今度の戦いの舞台はお城の中。床を落として敵をつぶすか，床ごと敵を落としてしまうか，とにかく全部をやっつけると面クリアである。

発売は6月下旬の予定。

X68000用　5"2HD版　5,300円(税別)

電波新聞社　☎03(3445)6111

遊んでばかりはいられないかな

		(前回順位)
1．	SX-WINDOW ver3.1	−
2．	餓狼伝説SPECIAL	4
3．	MUSIC SX-68K	−
4．	大魔界村	2
	スーパーリアル麻雀PⅣ	3
6．	アルゴスの戦士	6
	魔法大作戦	10
8．	スタークルーザーII	7
9．	F-Calc	−
10．	龍虎の拳	9
	サムライスピリッツ	8

先月号の「期待の新作ソフト」ではゲームソフトがベスト10を独占しましたが，今月は初登場の3本がシステムとツールで，また新たなひとつの波がきたようです。この集計は5月号のアンケートはがきによるものですが，連休の影響でいつもよりやや早めに行われています。つまり，これは5月号の発売後すぐに投函された読者の声なのですね。

3本はいずれも，Oh!Xの記事・広告に情報が掲載されたのは5月号が最初ですから，待ちのぞんでいたソフトのニュースに対し，みんながすぐに反応したということが表れています。最近シャープからも，たて続けに新作ソフトが発売されています。新機種の話がまだ出てこないのは寂しいことですが，待つあいだにソフトが充実していくのも嬉しいですよね。はやく，みんなの期待に応えてほしいものです。

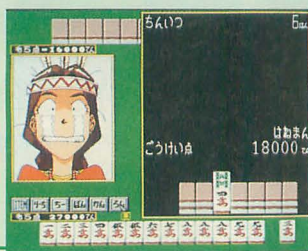
ここに挙がったゲームのなかには，まだ発売日などが決まっていないものもあります。こちらも開発の方たちはがんばっているそうですので，気長に待つとしましょうか。

それから，今回のランキングには入りませんでしたが，上で紹介している「Mr.Do！/Mr.Do！ VS UNICORNS」についてもすでにはがきは寄せられています。楽しみですね。

麻雀航海記

ゲーム内容についての新しい情報はまだお届けできないが，５月の発売に向けて開発は順調に進んでいるようである。

今回新しく入手した開発途中バージョンでは，麻雀以外の画面も見ることができた。世界地図やお買い物シーンなどから，ゲームの全体像を想像してほしい。登場キャラクターのグラフィックはいまの時点で入手できたのは残念ながら写真の少女だけだが，いろんなところに旅をするとのことなので，女の子もバラエティに富んでいるらしい。開発はアレックス。

X68000用　3.5/5″2HD版 5,800円(税込)
ブラザー工業(TAKERU)　☎052(824)2493

レッスルエンジェルス3

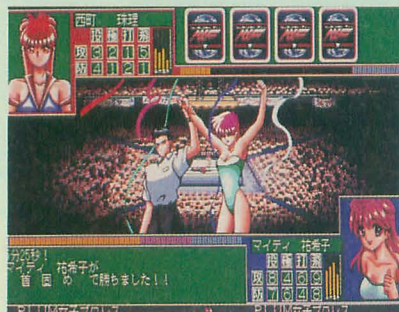
先ごろ発売された「レッスルエンジェルス２」に続いて，「レッスルエンジェルス３」の発売が決定した。

これまでの２作は，レスラー個人の闘いに焦点があてられた，チャンピオンを目指すタイプのゲームだった。しかし，今度の「３」ではまったく違っている。プレイヤー自身がプロレス団体を結成し，その新団体を経営していくのである。選べるレスラーは，新人や現役選手など約50名。彼女たちのうちの誰を仲間に加え，誰を敵に回すかはプレイヤーの自由である。

もちろん，これまでのカードバトルは健在だが，対戦画面は一新されそうだ。新たな技が追加されたり，ツープラトンでの攻撃もできたり，闘いもバリエーション豊かになるだろう。

とにかく，前２作とは趣を変えたこの「レッスルエンジェルス３」。いまから楽しみである。

発売時期は夏頃の予定。

X68000用　3.5/5″2HD版　価格未定
ブラザー工業(TAKERU)　☎052(824)2493

画面はPC-9801版です

CGAマガジン第5号完成!

プロジェクトチームDōGAより，ニュースをお届けします。

廃刊がウワサされていたCGAマガジンの第５号が，４月末に発行された模様です。内容は飛ぶもの特集ということで，いろいろな国，いろいろな時代の戦闘機などが収められておりますが，詳しい内容はこの原稿を書いている時点ではまだわかっておりません。しかしながら，これまで同様，かなり凝った形状データが掲載されていると予測されます。

CGAマガジン編集部では，廃刊のウワサを強く否定しており，５号の発行が遅れた理由を，CGAコンテスト開催のための計画的なものとコメントしております。しかし，関係筋の情報によると同編集部の労働力不足によるものであろうという見方もいぜん強くあります。

また編集部では，近く組織を変更して，第６号より内容も刷新する計画を進めています。しかし，完成時期およびその具体的な内容については，まだ正式なコメントはありません。今後の動きに注目していきたいと思います。

なお，CGAマガジンは，全国のソフト自動販売機TAKERUで入手できます。

以上，かまたがお伝えいたしました。

宝魔ハンターライム11

好評のライムシリーズも，いつの間にやらもう第11話だ。ホントにあと2つで終わっちゃうのぉ？ とファンをやきもきさせながら，肝心のライムたちは今日ものんきで元気な様子。

そこへやって来たのは，某芸能プロのマネージャ。スカウトかと思いきや，なんと，ライムたちを名(？)探偵と見込んでの仕事の依頼だという。そうか，こいつらいちおう「探偵」と称していたんだっけ。で，芸能プロならやはりお話はアイドルちゃん関係だよね。

アイドル＋事件＝ムフフな写真の流出，ってのが公式だけど，でもどーしてこんなものが，ってのがマネージャの悩みの種。そこで今回もライムちゃんは自ら身を呈して妖怪退治に乗り出すのでした。

期待の変身(変装？)コスチュームは，ココナちゃんとのアイドルデュオ。正義のためならカメラ小僧たちのえっちな視線にだって耐えるのだけど，ああ，それにつけてもバースの頼りなさ。そこで一計を案じたライムは……。ということで，あとはプレイしてのお楽しみね。

X68000用　3.5/5″2HD版　1,500円(税込)
ブラザー工業(TAKERU)　☎052(824)2493

画面はPC-9801版です

TAKERU名作文庫ソフト

2月号で紹介したブラザー工業(TAKERU)の，「名作文庫ソフト」。過去の名作を低価格で再販する企画である。この「名作文庫ソフト」に新たなX68000用ゲームソフトが加わった。

今回は光栄とズームの作品で，合わせて12タイトル。入手し損ねていた人には朗報だ。

発売開始は6月中旬の予定。タイトルと文庫価格は以下のとおり。( )内は元のパッケージ価格である。

光栄

タイトル	価格
信長の野望・戦国群勇伝	3,400円(9,800円)
信長の野望・武将風雲録	3,400円(9,800円)
三國志	5,200円(14,800円)
三國志II	5,200円(14,800円)
維新の嵐	3,400円(9,800円)
水滸伝・天命の誓い	3,400円(9,800円)
提督の決断	5,200円(14,800円)
大航海時代	3,400円(9,800円)
ランペルール	3,400円(9,800円)
リーディングカンパニー	4,500円(12,800円)

ズーム

タイトル	価格
ファランクス	2,500円(8,800円)
ジェノサイド2	2,500円(8,800円)

X68000用　3.5/5″2HD版
ブラザー工業(TAKERU)　☎052(824)2493

発売中のソフト

★F-Calc for x68k　クレスト 4/18
X68000用　3.5/5″2HD版 14,800円(税別)
ブラザー工業(TAKERU) 14,800円(税込)

★大魔界村　カプコン 4/22
X68000用　5″2HD版 9,800円(税別)

★あすか120% BURNING Fest.　ファミリーソフト 4/22
X68000用　5″2HD版 7,800円(税別)

★スーパーリアル麻雀PⅣ　ビング 4/27
X68000用　3.5/5″2HD版 12,800円(税別)

★アルゴスの戦士　電波新聞社 4/27
X68000用　5″2HD版 5,300円(税別)

★宝魔ハンターライム10　ブラザー工業(TAKERU) 5/10
X68000用　3.5/5″2HD版 1,500円(税込)

★MUSIC SX-68K　シャープ
X68000用　3.5/5″2HD版 38,000円(税別)

新作情報

★ザ・ワールド・オブ・X68000II　電波新聞社 5/下
X68000用　5″2HD版 価格未定

★麻雀航海記　ブラザー工業(TAKERU) 5/下
X68000用　3.5/5″2HD版 5,800円(税込)

★Mu-1 GS　サンワード 5/未
X68000用　5″2HD版 28,000円(税別)

★マージャンクエスト　SPS
X68000用　5″2HD版 価格未定

★宝魔ハンターライム11　ブラザー工業(TAKERU) 6/10
X68000用　3.5/5″2HD版 1,500円(税込)

★ロボスポーツ　イマジニア
X68000用　5″2HD版 価格未定

★Traum　象スタジオ
X68000用　5″2HD版 価格未定

★鮫！ 鮫！ 鮫！　KANEKO
X68000用　5″2HD版 価格未定

★達人　KANEKO
X68000用　5″2HD版 価格未定

★エアバスター　KANEKO
X68000用　5″2HD版 価格未定

★サバッシュII　ポプコムソフト/グローディア
X68000用　5″2HD版 価格未定

★麻雀悟空・天竺への道　シャノアール
X68000用　5″2HD版 9,800円(税別)

★スタークルーザーII　アルシスソフトウェア
X68000用　5″2HD版 価格未定

★魔法大作戦　EAビクター
X68000用　5″2HD版 価格未定

★龍虎の拳　魔法株式会社
X68000用　5″2HD版 価格未定

★餓狼伝説SPECIAL　魔法株式会社 7/未
X68000用　5″2HD版 価格未定

★地球防衛MIRACLE FORCE　カスタム
X68000用　5″2HD版 価格未定

★SX-WINDOW ver3.1システムキット　シャープ 5/下
X68000用　3.5/5″2HD版 22,800円(税別)

★XDTP SX-68K　シャープ 6/未
X68000用　3.5/5″2HD版 価格未定

★レッスルエンジェルス3　ブラザー工業(TAKERU)
X68000用　3.5/5″2HD版 価格未定

★宝魔ハンターライム12　ブラザー工業(TAKERU)
X68000用　3.5/5″2HD版 1,500円(税込)

★Mr.Do！/Mr.Do！ VS UNICORNS　電波新聞社 6/下
X68000用　5″2HD版 5,300円(税別)

おうちでPⅣ,グー!

Kiyose Eisuke
清瀬 栄介

あいかわらず,しゃべるは脱ぐわ,オマケ攻撃もすごいわで大騒ぎの「スーパーリアル麻雀PⅣ」。気になるアニメーションも思考ルーチンもアーケード版の評判どおり。忠実移植をビングに感謝しつつも3人娘をひん剝きましょう。

いまさら解説はいらないだろう。あの「スーパーリアル麻雀PⅣ(以下PⅣ)」の登場である。X68000版のデキはよい。気になっていたコンピュータの思考もアーケード版とほぼ同じだ。

残る問題はユーザー側の環境だけ。ディスク枚数が9枚という大迫力作品なので,ハードディスクの容量は十分に確保すること。フロッピーで遊ぶことは考えないほうがいい。なにせハードディスクで10Mバイトを使うソフトである。RAMも4Mバイトは確保しておきたい(動作は2Mバイトから OK)。

以上の環境を整えればもう大丈夫。コンピュータ麻雀界で圧倒的なブランド力を誇るPⅣが,ディスプレイに現れる。1993年のゲームだから,最近のX68000に移植されたアーケードゲームの中でも新しい部類だ。スーパーリアル麻雀がスピード移植されてしまうのだから,人間がんばればいいことがあるものである。がんばったのはビングさんだけど。

スーパーリアル麻雀伝説

最近ではゲームプログラムの規模が大きいのは当たり前で,規模の大きさをウリにしたゲームは家庭用でも業務用でもあまり見られない。が,メモリが高かった昔はデモやアニメーションに容量を割いているだけでウリになる時代があった。

X68000用 5"2HD9枚組 12,800円(税別)
ビング ☎03(3492)1079

なんで山中の一軒家が雀荘なの?

その中でもフル画面のアニメーションという剛速球1本で勝負していたのが,セタの「スーパーリアル麻雀PⅡ」である。アイテムはなし。スゴロクもどきのようなお遊びもなし。ラストチャンスなどのルール変更もいっさいなし。その代わり,登場する女の子がアニメビデオのように動き,しゃべり,そして脱ぐ。

これが効いた。プレイヤーは100円玉をザラザラ注ぎ込み,その人気は高いステイタスを生んだ。

加えて,麻雀ゲームのインチキくささを逆手にとり,異常なほどの強さをコンピュータにもたせたことが,このゲームの人気に拍車をかけた。作ったセタは,アニメーションを全部見せてしまうと飽きられると思って,簡単には見られないようにしたのだろうが,これが不思議に人気の秘密となってしまった。

ヘタすると1局目で「天和」。プレイヤーのしたことはお金を入れてスタートボタンを押しただけという……。PⅡはそのうち「しょう子にコテンパンにやられた自慢」というヘンなカルチャーまで生み出してしまった。

ここまで極端なことをしておきながら,プレイヤーに受け入れられてきたのは,ほかにはできない「フルアニメーション」を武器として有効に使ってきたからに違いない。プレイヤーは見事にハメられたわけだ。ハマッたこっちはバカみたいだが,幸せだからまあいいってことかな。

PⅣが自宅に!

このPⅣは,その伝説のPⅡから6年後,1993年に登場した4代目である。

その頃ボクの近所のゲームセンターでは,女の子を呼ぶことを意識し始めていて,ゲームセンターが改装されるたびに麻雀ゲームの数が減るというちょっとかわいそうな境遇だったが,それでも麻雀コーナーのあるところPⅣの姿があった。

PⅣでも「フルアニメーション」「コンピュータの強さ」の2大特長は健在だ。ただし,さすがにコンピュータの強さには多少遠慮がある。最初のうちはプレイヤーの配牌をよくしてくれるのだ。これはプレイヤーが勝ち続ける限り続くので,最初の1人をクリアするのはそんなに難しくない。

が,このツキは1回相手に上がられると逃げてしまう。たちまち配牌は最悪になる。ここから巻き返すのには相当な根気と集中力がいる。女の子が代わると点数も1000点に戻ってしまうので,配牌のいいうちに貯金をしておくこともできない。

極端なゲームバランスだが,ボクは認める。プレイにメリハリがあるからだ。気を抜くところ,抜けないところがあるからやってて楽しいのだ。こういう人為的な演出

3人娘がお出迎え

は”非”本格派麻雀ゲームならではの楽しさ。コンピュータ麻雀が本格派である必要はないと思ってるボクにはぴったりだ。

アニメはオープニングから炸裂する。雀荘のバイト募集にやってきたプレイヤーが，3人娘と対決するという設定だ。

香織「いらっしゃい。表の貼り紙を見てきたのね」

愛菜「それじゃ，まずあたしがお相手しまーす」

さすが9枚組。豪勢なアニメーションと音声である。画面もアーケード版と同じ高解像度モードだ。

1回相手に上がられるとクリアへの道は一気に遠くなる。つまり，どんなに安い手でもいいから早アガリを目指さねばならない。こっちのテンパイに対して無防備なのがまた，こういう麻雀ゲームのいいところ。1人目の愛菜はそんなに強くはないが，一度も負けられないと思うとそれなりにプレッシャーがかかる。ここで連チャンで倒せないようでは先の見込みはない。

「ロン！」

上がればそこは，PⅣフルアニメーションワールドだ。

恥ずかしそうに脱いでくれます

こっ，これがPⅣの真骨頂……

3人娘のキーワード

スカートを下ろす愛菜。ハッとこちらに気づいて，

「ヤダ，見ちゃだめ！」

んふふ。やはりアニメーションのデキは秀逸。簡単には真似できないものがあるな。ほかの脱ぎ麻雀と違って，このゲームだけはスキップしないで全部見てしまう。

女の子のかわいさは，スーパーリアル麻雀の生命線だ。アニメーションを見る気がしなけりゃ，遊ぶ意味がなくなる。

ボクの感想としては，PⅡほどではないにしろ，十分やる気になるグラフィックだと思う。原画を描いているのは，田中良氏。「幽☆遊☆白書」の劇場版の作画監督もしている方だそうである。やはり力を入れるところにしっかり力を入れているわけだ。

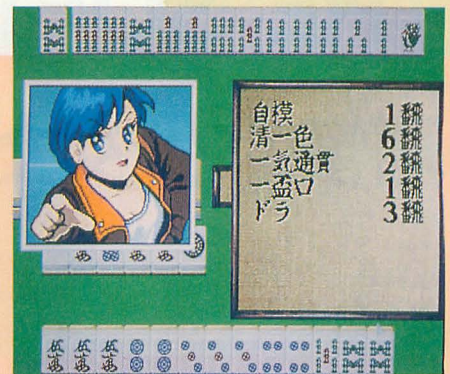
PⅡのしょう子は，アニメ系美少女の王道を行くスタイルだったが，PⅣでは女の子が3人いるということもあって，個性がつけてある。少女趣味の愛菜，ボーイッシュな悠，エレガント系の香織といった具合。

だけど，個性といったって，ボクはゲームセンターでは香織の顔なんて見たことない。悠だってやっと1枚脱がしたくらいなのに。X68000で遊べるようになって，ようやく3人いる意味があるようになったって感じだ。

キャラクターの人気を意識してか，オマケがこれまた怒濤の布陣である。絵ハガキ2枚，パンチングボウルにポスター引換券。半分キャラクターゲームにされちゃってるような気がするな。

こういうグッズって，マニアの熱狂的な要求に応えてポツポツ作るもんだと思うのだが，このPⅣでは最初から熱気でムンムンしたおマケが入っている。製作者側が先にあっちの世界にイッちゃってるような気がするのはボクだけか？

元祖で本家本元だ

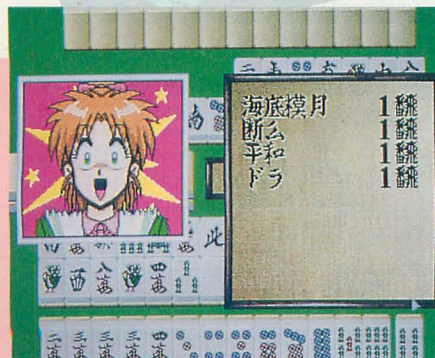
X68000の移植は前述のとおり。アーケード版と分けて考える必要がないくらい。大したBGMでもないのにMIDIに対応してしまったり，リプレイモードをつけたりと，サービスも満点である。スピードも10MHzで特に問題なし，X68000 XVIだと動きが軽いという印象さえあるぐらいだ。

麻雀ゲームは麻雀のシミュレーションならず。これをもっともよくわかっているのがスーパーリアル麻雀シリーズだと思う。インチキくさい配牌。とってつけたようなリーチ。やたら強いコンピュータ。

このPⅣを買う人は，そういう麻雀ゲームの魅力で買うか，愛菜・悠・香織のキャラクターの魅力で買うかどちらかだと思う。

どちらでもいいんだけど，ボクは個人的には「麻雀ゲームが好きだから」という理由にしておきたい。「香織が好きだから」とかいう理由で買うのには，なんか自分の中に許せないものがある。

ただのプレイヤーか，「スーパーリアル麻雀」マニアか。この線引きは，ソフトバンク刊の「スーパーリアル麻雀PⅣ 原画&設定資料集」を買うかどうかあたりにありそうだ。こういうモノを買っているようでは一般人とは呼べないぞ。

だけど，この資料集，結構売れてるらしい。なんでも3人のカラー原画とスーパーファミコン版の原画が収められている。さらに絵コンテや，ボツセリフ集，ボツ表情集も入っているのだ。

あ，一般人でなくてもいいような気がしてきた。

海底ツモ。ちょっとはツキがあるみたい

マニアもパンピーも買え！

X68000版「スーパーリアル麻雀PⅡ&PⅢ」ではコンピュータの思考ルーチンをマトモにしすぎた傾向があったが，このPⅣはゲームセンターの感覚をそのまま味あわせてくれる。原作を知ってる人なら一度は遊んでおきたいゲームだ。やっぱり家で聞くと音声もハッキリ聞こえるし，なによりX68000のディスプレイで高解像度のアニメーションがどんどん展開されるのには圧倒されるぞ。音楽もMIDIに対応しているし，手ぬかりはないって感じだね。

PⅡのしょう子，PⅢのカスミ＆ミキのほうが好きだったという人も，買っておいて損はないかも。なぜかというと（秘密）。

なお，今回はサンプル段階で評価したため，「麻雀バトルモード」などのパソコン版ならではの機能を試すことができなかった。機会があれば，追加レビューを行いたい。

総合評価	0 — 5 — 10
グラフィック	★★★★★★★★★★
音楽	★★★★★★★★
ゲームバランス	★★★★★★★★★★
オマケ	★★★★★★★
熱中度	★★★★★★★★★★

THE SOFTOUCH

●あすか120%BURNING Fest.

可憐な少女たちの汗と野望

Taki Yasushi
瀧　康史

ロボットの次はお嬢様の登場です。生きてくことは闘いなのさ。いずれ劣らぬ6人の美少女たちも，その運命からは逃れられないのでした。試験管やポンポン，ボールが飛び交う舞台は女子高。お嬢様歩きもちょっと可愛いぞ。

気分は120%

私立繚乱女学院，通称「乱女」。創立以来，政・財界や文壇，芸林といった各方面に多くの著名人を輩出して，わずか30年にして「名門の誉れ」ここに極まった感のあるいわゆるお嬢様学校である。創立者・新堂源一郎学院長をはじめとする志高き一流の教師陣が，文武両道をモットーとして，次代を担う淑女の育成をめざしていた。

その繚乱女学院高等部名物「部対抗予算争奪戦メガファイトトーナメント」(略して「メガファイト」)は，毎年秋に行われる文化祭「百花祭」で恒例となった一大イベントである。それぞれの部の代表が対戦し，優勝すれば部費が増えて1年間リッチに部活動ができるという，どう見ても運動部に有利なシステムだ。

主人公は本田飛鳥。「メガファイト」制覇を夢見る文化部の期待を一身に背負ったファイターだ。ちなみに化学部。

「マッドストーカーX68」のスレイブギアが可愛らしい女の子になったのかなぁ……なんて甘いことを考えていたら，そこには全然違う世界が待っていたのだ。

実は入門格闘ゲーム？

このゲームは売れるだろう。

画面からも察せられるキャラゲーとしての風格，初心者でも遊べる対戦格闘ゲームとしての品質，さらにその上の，上級者のための難易度最高格闘対戦ゲームの地位。

初めは，現在「マッドストーカーX68」が売れているときに，なぜ同系統らしきゲームを発売するのか，いささか不思議ではあった。ところが，このゲーム，やりこめばやりこむほど別物だという認識が強くなった。そして，「ストII」や「餓狼伝説」シリーズにはない新しい試み，知らぬ間に身についていた対戦ゲームの常識を覆すようなゲームシステムに気がついたのだ。

現在の対戦格闘ゲームの多くは，もはや新しいプレイヤーをとらえる魅力はなく，古いプレイヤーの心をつかみ続けることに焦点が置かれているように見える。難しくなりすぎて初心者には入り込みにくくなっているのだ。たとえコンピュータが相手でも，まったく勝てなければつまらない。

この「あすか120%BURNING Fest.」(以下「あすか」)は，EASYモードにすればかなり勝ち進むことができる。連射スティックを使ってとにかく技を出しまくれば勝ってエンディングまで進めるという，まさに格闘ゲーム入門の要素をもっている。格闘ゲームをやったことがない人，面白そうだとは思いながらも難易度などで敬遠していた人には，ぜひチャレンジしてほしい。

本当は熱い，格闘ゲーム！

一般に，入門向けに難易度を設定しているものは上級者には面白くない場合が多い。しかし，「あすか」の対CPU戦は入門向けだが(ただし，HARDモードはそうでもない)，人間，特に上級者同士の闘いになると信じられないほどゲームフェイスが変わる。

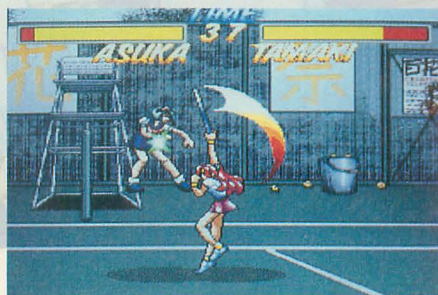
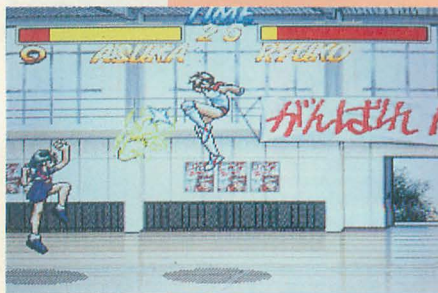
ほかの対戦ゲームにはない新しい試み，これが「あすか」の大きな魅力だろう。

現在，世に出回っている大半の対戦ゲーム(「バーチャファイター」は例外)には，かなり共通した「暗黙の了解」がある。例を挙げると，空中で一度攻撃を食らうと無敵になるとか，倒れる技を食らうと倒れるまで無敵だとか，空中では防御ができないとかいうものだ。そして，これらの暗黙の了解から「跳ばせて落とす」という戦法が生まれてくる。ところが，「あすか」の場合は空中で防御ができるので，このやり方は通用しない。

たとえば，本田飛鳥が遠心破砕拳(飛び道具)を撃ち，対戦相手である新堂環が，それを避けるためにジャンプしたとしよう。飛鳥は環の落ち際にダッシュ技などをかけようとする。「ストII」と同じシステムなら環は技を食らうだけだが，このゲームの場合，環には次の選択肢がある。

1) 2回目のジャンプ
2) ダッシュ技をただの大叩きによって空中で相殺して，
　① もう一度ジャンプ
　② もう一度叩く
3)空中防御(大ボタン＋小ボタン)
4)素直に食らう

と，簡単に思いつくだけでもこれだけある。

X68000用　5"2HD版　7,800円(税別)
ファミリーソフト　☎03(3924)5435

学院長令嬢は優雅にテニス

そーれっ！のかけ声でボールが飛んでくる

仮に飛鳥がこれらを予想して，昇り蹴りをしたとしよう(大なら2段攻撃)。この場合でも環には次のような選択の余地がある。
1) 小，大攻撃における相殺
相殺後，飛鳥と環両方に次の権利がある。
① 相殺後のジャンプ(逃げor攻め)
② 空中防御
③ さらに攻撃
④ 空中投げ
⑤ 空中必殺技
(ただし全キャラ中，環だけにはない)
2) 2段ジャンプ
(逃げorタイミングを崩すための攻め)
3) 空中投げ
4) 空中防御
5) 素直に食らう

素直に当たってしまった場合，環が吹き飛ぶため(その分，長く空中にいる)，飛鳥は早く地上に降りる。空中で攻撃を食らっても，このゲームでは無敵にならない。

ということは飛鳥には，もう一度ジャンプして攻撃をしたり，相手の落ち際に何らかの攻撃をするという選択が待っていることになる。環は上に書いたような空中防御などの対処をしなければ，空中から落ちずに延々と攻撃を食らってしまうのだ。

このように，「あすか」にはいままでの格闘ゲームが築いた「暗黙の了解」を裏返すいろいろな試みがあり，プレイヤーに息つく間を与えない。これに慣れるとほかのゲームがうざったくて，かったるく感じられてしまうほどだ。

ここで，鋭い読者なら「相殺し合うゲームだったら，多段攻撃をしかけたほうが勝ちじゃないか」と思うだろう。ところが，さすが「マッドストーカーX68」を開発したフィルインカフェ，その点もしっかりと考えられている。

それが，受け身，返し技だ。

必殺技には，相手を水平に飛ばしたり(ダッシュ技など)，垂直に飛ばしたり(環のロブ落としなど)する技がある。これらを食らったとき，飛ばされた方向と反対にスティックを2回倒せば，受け身がとれる。

また，連続技を含む猛攻ラッシュを食らっているとしよう。こういうときはたいてい地上にいる。ここで，立ち防御をしながら強と弱ボタンを同時に押せば，返し技を出すことができるのだ。これは具体的には，立ち大攻撃を当てるというもの。間合いが近ければ，そのあと投げることもできる。立ち大攻撃は当たると敵が少し飛ぶので，態勢逆転ができるというわけだ。この返し技は，敵の連続技を食らっている最中にも

お気に入りのアノ娘で対戦しよう

応援部だって自ら闘うのだ

出せるので，従来のゲームのように，技を食らっているのを指をくわえて見ている必要がない。

これでは待ちにはしるのでは，と思うかもしれないが，そういうわけでもない。返し技は立ち防御の最中にしか出せないし，リーチの長い技や飛び道具は返し技をしても当たらない。それどころか，立ち大攻撃を空振りする分，隙ができるし，立ち防御の状態は足払いで簡単に崩される。飛び道具かスライディングのような高速な足払い，リーチの長い技のどれかは必ずあるので，返し技待ちではまることはないだろう。

しゃがみ防御と返し技を繰り返すような待ちをされても，大攻撃で体力を削ることができるので，しゃがみ大や，必殺技を使ってどんどん削ればよい。また，返し技待ちをしていても，必ず防御グラフィックに変わるので，すかして投げてもいいだろう。もっとも，投げのダメージは少なめだ。

あと，特徴があるのが小攻撃と飛び道具。

飛び道具はどんな技でも相殺されるので，ピンタなど小攻撃で相殺できる。小は連射できるため，無意味な飛び道具は放てなくなるし，また，連続できる分，多段連続技を無理につなげるような小攻撃は，ダメージは本当に少ない。

どうだろう。練りこまれているとは思わないか？

まとめ

ゲームスピードはターボモードにすると錯乱するほど速いゲームに変わる。ただし10MHz機ではそれほど速くはならない。すでに発売されているFM TOWNS版とは，まったく違うゲームとなっているようだ。

不満をいえば，グラフィックにあまり肉感がない。「ストII」の春麗の大腿4頭筋やキャミィの上腕二頭筋と三角筋，あるいは「餓狼伝説2」の不知火舞の，むっちりした女らしい肉感。見えそうで見えない転んだときのあの姿勢……(私を含め，あぁぱぁな男どもを惹きつける何かがあるだろ？)。

ゲーム全体については，あまりにも新しいタイプの対戦格闘ゲームなので判断に迷った。対CPU戦は簡単で初心者にもかなり遊べる割に，信じられないほど奥が深い。いうならば，初めて「ストII」をしたときの感触に似ている。まだ何かできるのではないか，という感覚がゲーマーの心を引き立てる。

「マッドストーカーX68」とは明らかに違う路線の格闘ゲームを，こんなにも短期間に仕上げてくれたスタッフには，ただ驚くばかりである。

追伸：女の子たちはよく喋るからね。

女の子だからって，ナメたら痛いかも

最初にプレイしたときには，対CPU戦はへらへらしているうちに終わってしまった。2作目というのは失敗しがち(？)な傾向から，「マッドストーカーX68」はよかったけどこのゲームは駄目か，とまで一瞬思った。正直，最初は奥の深さは見えず，表面だけで判断してしまうところであった。

ゲームはMIDI対応。TG-100なんかにも対応していてビックリした(その代わりにMT-32がない)。

フロッピーディスクはデータを圧縮してあるため2枚。その分ロード(というか展開)がむちゃくちゃ遅いが，ハードディスクに入れるときには，圧縮を解いてインストールできる。キーディスクが必要なのが非常に残念だが，待ち時間がぐっと減るのでよい。付記すれば，ちょっと，10Mショックかなぁ……なんて。ハードディスク＋16MHzは必需品かも。

そういえば，「マッドストーカー」のPCエンジン版が出るようだ。まわりにエンジンユーザーがいたら，絶対に買わせて対戦に走ろうぜ！操作法など，X680x0版で不満だったところが直ってたら，私も買おうかな。

それにしても，あの同人誌のようなマニュアルはやっぱり，デフォルトなんだろうか？

総合評価 (0 — 5 — 10)

項目	評価
ゲーム性	★★★★★★★★★
グラフィック	★★★
音楽	★★★★★★★★
効果音	★★★★★★★★
トーク	★★★★★★★
お買い得度	★★★★★★★★★

THE SOFTOUCH

●ジオグラフシール

ポリゴンの細道(越後編)

Yaegaki Nachi
八重垣 那智

発売後，大好評の「ジオグラフシール」。読者からの評価はとても高く，技術だけではなく，ゲームデザインも評価され作品自体の質の高さを物語っています。今回は，そのデザイン面を中心に八重垣氏が徹底評価を下します。

ゲームは趣味のものだから，もちろん誰しも好き嫌いはある。人一倍好みが偏屈なおかげかどうか知らないが，最近，どうも嫌いなゲームが多くなった。体を動かす快感が伝わらない多くのスポーツゲームは前から大嫌いだし，距離感がピンとこない3Dカーレースは画面がキレイなのでせいぜい見るだけがいいところ。対戦格闘ゲームも，結局，小器用にレバーとボタンを入力するだけの作業というものに煮詰まってしまってからは，ゲームと呼ぶのも嫌になってしまった。

確かにどのゲームでも，ゲームであるがゆえに，プレイヤーが慣れたりして作業化していく部分があるのだが，それをあえて意識させないことが，ゲームのリアリティというものではないのだろうか。

操れ！鋼鉄の跳ね馬を

いつになくくどい前置きだが，今回は2カ月にわたって紹介したエグザクトの「ジオグラフシール」の徹底評価である。ゲーム自体に関しての説明は，ほとんど不要だと思われるが，やはりそのあたりから押さえていくことにしよう。

ゲームのスタイルは，3Dのコックピット視点から見るタイプのシューティングゲームである。画面は基本的にポリゴンで作ったものが動き，存在している。通常は見えない自機は，4種類の武器を装備可能な歩行戦車だ。歩くからにはゲーム中ではフィールドを自由に動き，敵を叩き潰していく。

敵を踏みつける爽快感！　たまらないね

この戦場での自由な動きが可能という点が「ジオグラフシール」最大の特徴であり，平面の移動だけにとどまらず，ジャンプによる重力を意識した移動を盛り込むことで，空間を感じさせ，距離と高さが体に伝わるという独特なシステムを構築している。この動きがリアルタイムで計算されるポリゴンによって表示されることに，このゲームのすべてが凝縮されているといっても，あながち間違いではないだろう。目の前にディスプレイの表面ではなく，奥行きのある空間が広がっているのである。

これらの具体的な操作は，いつものレバー1つにボタン2個で実現されている。特に多段ジャンプが4段まで可能になっている点が，一瞬ショットボタンの存在を忘れそうになるほど虜になる踏みつけ攻撃の快感を提供している。ただゲームとしてのメリハリをつけるためか，実際にはこの空間多段ジャンプを駆使する面ばかりでゲームが構成されているわけではない。中には空中戦や迷路の面もある。

しかし，ジャンプする感覚に惚れた身としては，こういう構成が，もったいないと思ってしまったりする。もっとジャンプを，と声を大にしていいたい。

X68000用　5″2HD版　9,800円(税別)
エグザクト　☎025(284)7304

戦え！空間の中で

ゲームの展開は全8ステージで，基本的に敵の防衛網をくぐり抜け，そのステージのボスを破壊すればステージクリアという仕組みになっている。ボスにいくまでは，所定のターゲットを破壊しないとゲートが開かないなどのノルマが課せられていたり，3Dの空中戦をしてみたり，迷路を突破させられたりと，いろいろな趣向が凝らされている。どれもなかなか手応えがあって一筋縄ではいかないのが特徴だ。

では，いままでやってなかったので，一応ゲーム内のステージを紹介しておこう。基本的にその面の流れは最初に出てくるインフォメーション画面で押さえられるので，最初のうちはうかつにボタンに触らないで見ているほうがいいだろう(ただ，最後のほうではインフォメーションがなくなってしまうので注意)。

●1面：砂漠

ここのターゲットは少ないので，さっさと料理しよう。ボスも動かないが，踏みつけ防止の傘がついているのに注目。弱点が高いところにあるのだから，自分も高いところに昇って攻撃することに気づかないとハマる(苦笑)。

●2面：都市

ターゲットは自機のような歩行戦車。手

ちょと情報不足のミッション説明。かっこいいけどね

軽に踏んでも射ち殺してもいい。ボスの動きをよく見て，踏みつけも忘れずに使って倒せ。

●**3面：海上(空中戦)**

うってかわって強制スクロールの3Dシューティング面。護衛つきのアイテムキャリアは逃さず倒したい。

●**4面：迷路**

天井があるのでジャンプ(回避)不能，しかも見通しが悪いので拷問状態。さらにターゲットはダミー。シャッターを開けるのを優先して戦うようにするが，道は丸暗記しないとダメ。

●**5面：ダクト**

またまた強制スクロール，棒にぶつかっていると，ボスでシールドが寂しくなるので避けを重視。ボスは長期戦なので落ち着いていこう。

●**6面：基地**

中ボスが意外に強い。ボスはどこを壊すかがわかりにくいが，とりあえずジャンプして射ちながら踏んでいくしかないようだ。

●**7面：タワー**

塔のコアを破壊する面だが，落ちるとダメージを受けるので慎重にいきたい。しかもメインターゲットは変な板で自機を落とそうとするので，ジャンプでかわしながら攻撃しよう。

●**8面：宇宙**

なにをやりたいかは一目瞭然の対決空中戦。思ったほど難しくないかも。勝てば真の勝利が待っている。

ターゲットを倒す必要のない迷路面。バグか？

この石板に落とされるとちょっと痛いぞ

進め! 自分の腕を信じて

技術的にも，視覚的にもハイレベルなゲームなのだが，いくつか無視できない問題を構造的に抱えている。特に，それがプレイヤーに対して不親切であるという印象を与えているように感じるところがあるので，ここでハッキリ指摘しておきたい。

まずシューティングゲームの体裁を取っているのに，ショットの発射に制限があるという点である。設定的にはオーバーヒートなどと説明がつくところなのだろうが，はっきりいって無駄な仕様である。シューティングで自由気ままに攻撃できないというのは，息ができないのと同じくらい苦しいことなのだと考えていいだろう。

ボタンを頻繁に押したり離したりを繰り返していると，その気がないのに武器チェンジしたりするので，攻撃しながら敵のリアクションを見るより，ゲージを気にするようになる。かといって武器チェンジをメニューでやるようでは，ゲームの流れに水を差すので，これまたテンションが下がってしまう。こんなようでは爽快感とはまったく逆の方向に離れていくように思われてしかたがない。

次にほとんど役に立たないサブ画面のマップである。これは，原則的に迷ったときに参照するものにもかかわらず，表示領域が狭く，自機の方角も把握しにくい。そもそも現在の方向すら，このマップでないとわからないのも問題だし，最も頼りたい場合のターゲットを見失ったときに慌ててマップを見ても，全体マップ上での現在位置すらわかりにくいので，途方に暮れてしまうこと間違いなし。これは最も深刻な問題だろう。そもそもこういった移動の自由度の高いゲームにおいて，常にフィールド全体を把握できるなんらかの情報が常駐していないのは，戦略性をプレイに要求しつつ，その材料を提供していないという意味で，極めて不親切な設計であるといえる(その点「グラナダ」や「ボスコニアン」のマップ表示のほうが，はるかに役立つ)。

そして一番の問題は，そのデザインの不統一感である。特に明らかな示唆もなしにゲームの進行維持にかかわる部分が変わってしまう(特に4面のターゲットは壊す必要がないなど)のはいただけない。目先が変わっているといえば聞こえがいいかもしれないが，各面のシステムがバラバラで，締まりがない印象が先に立った。

しかも，そういった情報をゲームからハッキリとした表示や文章で得られないのも，これまた不親切といわざるを得ない。もうすこしゲーム全体でプレイヤーが身につけた操作や知識から考え，高度な操作やプレイに進んでいくような要素があるべきだと，痛感した。

ゲームが難しいというのは，なにをするかがわかっていて，それができないということである。ゲームをするために必要な情報すら提示しないのに，それを自分で「難しさ」などといっているのをみると，これはかなり問題だと感じてしまう。

しかし，これらのことを現実的には気にせずに，苦もなく楽しめる人がいるのもまた事実である。このゲームの価値を一方的に否定しようというつもりは毛頭ないが，これだけ素晴らしい表現や世界を，閉じた方向にもっていってしまうというのは，少なくとも私個人にとって「不親切」以外のなにものでもないと感じたのである。

こいつはアイテムをもっているので逃すな！

カッコよさと遊びやすさ

以上，とにかくプレイ中ものすごく気に障ることを書きました。これも，できるはずなのにあえてやっていないように感じるからこその意見なのですね。私としては，いいかげん「技術のエグザクト」などと書きたくはないのです。

実際，この表現力を支える技術や手法といったものに，よく練られた確かなものを感じ，驚いたり，感心することは事実です。でもそれだけでは，人は満足できないのだと思います。でも余計なことをすると，「過ぎたるは及ばざるが如し」などと書きたくなるから，実はちょっと足りないくらいのほうがいいのかもしれないな，うんうん。

総評

	評価 (0–5–10)
ゲーム性	★★★★★
技術	★★★★★★★★★★
サウンド	★★★★★★★★★
グラフィック	★★★★★★★★★
3Dのワビサビ	★★★★★★★★★★★

特別編11

己れの道を行くのさ，フッ RYU

Sudo Yoshimasa 須藤 芳政

頭はキチンと散髪しているのになぜか顎に不精髭を生やしているリュウ。金色に髪を染めているのに眉毛だけ黒いという，歌舞伎町あたりではいまや天然記念物となってしまったロッカーのようなケンへの密かな対抗意識か？　1P側でゲームを開始するとリュウにカーソルが合っているにもかかわらずスティックを1つ下へ入れられてしまうことが多いことへの嫉妬の表現か？

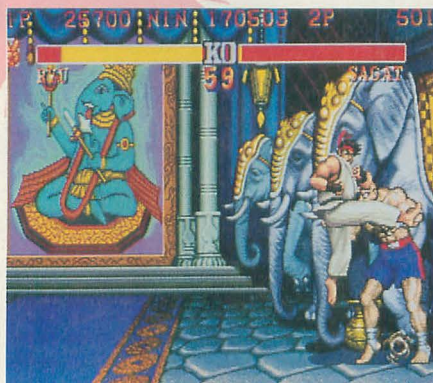
画面狭しと跳び回るケンに対して，リュウは地味な存在である。ノロノロと回る竜巻旋風脚は「竜巻」というよりむしろ「のり巻」であり，血の気の多いプレイヤーにはストレスをもたらす原因となる。昇龍拳も上昇下降がノロいのでケンのように「ドラゴンダンスを踊ってちょっとした余興を」なんて気の利いたことができない。

「リュウって，ケンみたいに芸とかできないのー？」

「無器用ですから……」

おお！　実はリュウが憧れている人は高○健さんだったのだ。おそらく落ち着きのない金髪野郎が憧れのあの人と同じ「ケン」であることを許せないので「自分はあのような闘いはしない！」と心に誓っているのだと思われる。

波動拳！ 波動拳！ 波動拳！

リュウの得意技は波動拳だ。技のモーションがケンより速く，跳び込み後にしゃがみ中キックなんかをからめて波動拳へつないでいけば，相手を端へ追い詰めて「鳥かご状態」へもっていける。が，しかし！　画面端っこへ追い詰めて「波動拳！　波動拳！　波動拳！……」相手が跳んだら「昇龍拳！」という作業。本当にただの「作業」にすぎない。

うおー！　臭いをかぐなー！

風呂場でデュエットソングを歌う

こんな戦法で連勝を続けたからといって「ちょっち俺のリュウ最強かもしんないっしぃ～」などと浮かれていると必ず誰かにハメ殺される運命だ。

こうならないためには竜巻旋風脚を活用する。リュウの竜巻はノロノロしてはいるが，当たれば相手が転倒してしまう優れモノ。何よりもこの技を鳥かご作業中に使用することによって，自分がハメ野郎でないと対戦相手に錯覚させて，胸を張ってハメることができるのだ(程度によるが)。

「うあああ……」You Win！

リュウの昇龍拳はほぼ真上にしか飛ばないし，サガットのアッパーカットのように破壊力が優れているわけでもないので，大昇龍は相手の体力が1ドットになったときの削り殺しをハデに決めるぐらいにしか，私は使っていない。大昇龍だと相手が転ばないので，アッパー大昇龍なんかをザンギにやったりしたら，待っているのは「フウッ！　グルグル」だ。小昇龍は滞空時間が長いこともあって相手の気合弾をかわすのに便利なことは皆さんご承知の通り。

相手が離れているときに波動拳を撃ち，相手が跳び越えてきたらすかさず上り大蹴りを食らわせる。これを2回ほどやれば，対戦相手を不快な気分にさせることができるだろう。相手があまりにも弱いなら，密着して中キックで膝蹴りするのもグー！　相手が気絶しているときには，いったん画面の端へ離れていき，相手が「おお，待ってくれるのか？　優しいー！」と思った瞬間，大竜巻で跳んでいってボコっと蹴るのがオススメ。めくり大キックを入れてしゃがみ小キックを2回ほど入れたら，再び跳び上がってめくり大キック＋しゃがみ小キックを繰り返す……これが全部入ったら対戦相手の精神はズタズタだ。

でも，よいこはマネしちゃダメだよ。

人気者のお前なんかキライだっ。バキ！

リュウちゃんキビシー！

特別編12

帝王流殺しのアンサンブル

VEGA

Hino Asao 日野 麻也

ベガはその顔からも想像できるように，かなりの強さを誇る。ほかのキャラで腕を上げても，その日に始めたばかりの完全トーシロベガにパーフェクト負け，なんて話も聞く。そう，とにかく強いのだ。使えばわかる。サイコクラッシャーを乱発すれば，あっという間にエンディングまでいけるはずだ。あまりに簡単で面白みはないが，勝ちたいだけならOK。これから書く珠玉のコンビネーションや帝王たらんとする者の心得は必要ない。ガンガン，サイコクラッシャーをキメてくれ。しかし，巷のゲーセンでは注意しよう(まあ現在は,ダッシュはほとんどないと思うが)。対戦でそんなことをすれば背後から刺されても文句はいえないぞ。それで許されるのはガキのうちだけ(本当は許されないが)。良識ある大人ならそんなことはしないはずだ。まあいいや。能書きは無用。帝王にふさわしい一流の生き方をお伝えしよう。

王者の美意識は優雅なる冷酷

サイコクラッシャーを使わずにいかにシブくキメるか。これが最大のテーマ。このシブさの演出には，ダブルニープレスを軸として，ヘッドバントプレス，跳び蹴り，跳びパンチをフルに使う。もちろん，それ以外の技もあなどれない。たとえば立ち大キック。一見シンプルなまわし蹴りだが，見かけと裏腹に異常なすばやさと射程，威力を兼ね備える。中間距離で飛び道具を撃とうとした相手に対して猛威をふるう。「波動拳」の「はど」まで聞いて出せば悪くても相打ちで，与えるダメージは桁違い。これを使いこなすのが大人への第一歩だ。

プレリュードはあくまでも優しく奏でよう

隣りのリュウに対抗して風呂場でペアダンス

そして，大人とお子様の決定的な違いは，ダブルニープレスである。これこそベガの真の恐怖の技だ。溜め時間が多少長めだが，そんなことはさして問題ではない。あらゆる溜め時間はこの技のためにある。たとえばダブルニープレス→しゃがみ中キック→立ち小キック→ダブルニープレス……などでお手軽に死刑執行できる(実はダブルニーハメと呼ばれて人々を震撼させた技である)。相手はガードしようとしまいと関係なく，ひたすら嵐が止むのを待つだけ，すなわち殺られるだけだ。返せないこともないが，下手に手を出せば死期が早まる。これまたサイコよろしく禁じ手とする。これは，単独使用でもかなりの威嚇となるが，連続技を織り交ぜてこそ真価を発揮する。いろいろあるなかで最高のものをお届けしよう。凄絶のひと言に尽きる技，それは，めくり大キック→しゃがみ大パンチ×4→立ち小パンチキャンセル→ダブルニー(小)→しゃがみ中パンチ→立ち大キックの10段攻撃。男なら一度はキメねばなるまい。ザンギあたりにデクになってもらおう。マイナーバージョンもいろいろある。跳び大パンチ→しゃがみ小パンチ×2→立ち小パンチキャンセル→ダブルニー(小)→立ちキックなどだが，大切なのは立ち小パンチでキャンセルをかけること。これ以外にはしゃがみ小パンチしかキャンセルできない。それとダブルニーを小キックで出すこと。まともにキマれば相手は生きていないだろう。そう

必殺技は使わないのも美学なのである

そう，あまり知られてないようだが，しゃがみ中パンチは2発連続で入る。これはベガに限らず，多くのキャラで可能。ガイルでしゃがみ中パンチ→しゃがみ中パンチキャンセル→大サマーソルトなんてのもOK。ベガなら，めくり大キック→しゃがみ中パンチ×2なんかで十分ピヨリも狙えるぞ。

これだけでも十分強いが，さらに上を目指す貴兄にはヘッドバントプレスをお勧めする。飛び道具を撃たれたときに使うのが主だが，何でもないときに急に出すのもトリッキーでなかなかデリシャス。そしてその後のサマーソルトスカルダイバーの使い方でアマとの差が出る。遠くに行くと見せかけて急に近づいて攻撃したり，めくり位置に入れてガードさせないようにしたり。とにかくどんどん出して蠅のようにブンブン飛びまわり，相手にストレスを与えるのだ。そのうちキレて自滅してくれるだろう。

こんなベガにも弱点はある。それは対空技がないことだ。いや，あるにはあるが極めて弱いのだ。まずはサイコクラッシャー。早めに出して自分の足が相手に当たるようにするのだが，溜まってなければ使えない。ほかにはしゃがみ大パンチがまあまあ使えるかな。間に合うなら跳び蹴りも有効。このくらい。ダブルニーもいいけど，これも溜め時間の関係で使いにくいはずだ。

要は，守りに転ずると弱くなる。サイコクラッシャーを使いたくなるだろうが，帝王のプライドでグッと我慢。攻めまくれば問題はない。技の威力が大きいので相打ちだろうと強気でガンガンいくべし。フィニッシュをしゃがみ大キックなんかでキメると，ビンゴッ！　って感じでモアベター。

しかし，対戦ではほどほどにね。友人関係には気をつけたほうがいいですぞ。

がんばればなんとかなります

Yokouchi Takeshi
横内 威至

ひたすら難易度の高い「大魔界村」の攻略に挑んだ横内氏は玉砕状態。しかし「がんばればなんとかなる」を合言葉にひたすら努力しよう。エンディングにたどりついたときには，画面が涙でかすんでいるかもしれませんが。

カプコンの新作「大魔界村」ですが，かなり極悪な難易度を誇っている，といわれています。俺としては「魔界村」のほうが好みなんですけど，見た目カッチョいいのはこっち（ちょっとくやしい)。

結構難しいので，ちょっとプレイしただけであきらめた人も多いかな。でも，がんばろう。なにごとも我慢が肝心です。楽しまなければだめでしょう。拷問的とはいっても拷問を楽しむ人もいるんです。だから前向きになればきっとこのゲームも楽しめるはずなんです。たぶん。では「大魔界村」を楽しんでみることにします。

豪勢なゴミ箱について

一応，宝箱には各種のアイテム，いわゆる宝物が入っているらしいのですが，ゴミのほうが多いといえるでしょう。ゴミのなかには迷惑なオヤジが埋蔵されているし，好ましくありません。しかも，オヤジの魔法を食らうとアーサーがアヒルになったり爺さんになったりします。開けるときには細心の注意が必要です。

で，この宝箱の中身は，出現する順番によってある一定のパターンに沿っています（死んだら出現順番はリセット)。

まずアーサーが鎧を着ているときの出現順序から。1つ目はオヤジ，2つ目は金の鎧，次は武器，そして再び金の鎧。アーサーがセクシー状態のときはかなり拷問です。1つ目はオヤジ，2つ目もオヤジ，3つ目は武器，そしてやっと4つ目で鎧です。金の鎧を着ているときは，1つ目がオヤジ，2つ目が武器。

ということなのですが，何個目，というのは宝箱を開けた数ではなくて出現した数です。いちいち宝箱の出現個数をカウントしていけば，無駄な宝箱を開けずにすみますが，そんなことに頭を使っている余裕などないかもしれません。宝箱は，変なところで跳ねたりすると出たりもするので，覚えられる範囲で覚えておくのもいいでしょう。

また，裸で金の鎧を着てもノーマルの鎧になることに注意。むしろ金の鎧は脱がされたときの保険としたほうが賢い使い方です。どうせ魔法はまったく役立たないんですから。しょせんカッコだけ。

X68000用　5"2HD版　9,800円(税別)
カプコン

自虐的パワーアップ

今度は，武器のおさらいをしてみましょう。ノーマルの槍は普通に使えます。上打ち，ジャンプすれば下撃ちなんかもできて2連射もできます。ほとんどこれで十分。

そして出たら取るべきなのがナイフです。まっとうな人が考えると槍のほうが強そうなのですが，ナイフのほうが素早くて，しかも3連射できるのでかなりお得。これでいくのがベスト。

あとの武器は雑魚，ほとんどカスばかりという情けないラインナップ。たまに使えるけど，それ以外は拷問としかいいようがありません。

まずは斧。これを取ってしまったら，もう希望は捨てよう。さらに素晴らしいのが剣。取ったらリセットするか，気晴らしに街にでも遊びにいきましょう。タイマツは使えそうで使えません。レッドアリーマーなんかに使えるときもあるけど，上にほとんど飛ばない，燃えてる間に連射できなくて歯がゆい，遠くに飛ばないなどで結構イヤ。好みの問題もあるかも。そして円盤。地面に沿って飛ぶのはいいんだけど，いってほしくないところに転がっていったりして困ります。しかも，いまいち使えるところがありません。音が気持ちいいと感じたら使いましょう。

まあ，もはや通常プレイで満足できなくなった人は，わざと使えない武器を取って修羅の道を歩むのもいいでしょう。ただ，そんなことに労力を使うよりは，人としてもっとすべきことがあると思います。

難しい1面

八重垣氏いわく，1面が最も難しいらしい。ここでコツをつかめばあとはたいした壁はない，らしい。確かにそんな感じの作りとなっています。全体的に，基本として厳守すべきことがいくつかあるので覚えておきましょう。

まず，跳ねまわらないこと。跳ねて進むとちょうど着地点にうまく敵が現れ，とても痛いです。そして，むやみに宝箱(ゴミ箱？)を開けないこと。最初の説明のとおりです。で，敵をむやみに倒さないこと。下手な位置で殺すと結構いやらしい武器を落としていくことがあります。ジャンプ先に斧とか落とされるとアツいです。そうなったらあきらめて死ぬのも美しいでしょう。

ということで1面だけど，敵がHなところから現れるのに注意し，また上から降っ

こいつはパターンがあるから楽勝さ

てくるのに気をつければたぶん大丈夫。走ってくるカマ男には気をつけよう。「走ってくるカマ男」と聞いてムラムラしてしまうあなたは，即座にゲームをやめて某２丁目に走るべきです。俺はムラムラしないから大丈夫だけど。

ということで，最高に難しい後半です。俺はあまりの苛立ちと屈辱に耐えられず，カプコンさんに牛の生首を贈りました（ウソ)。ところで，このゲームをするときには，一応，キレると危ないのであと腐れないようにアルコール類やドラッグ類を用意すると，X68000を壊したりはしなくなるかもしれません。ちょっとした生活の知恵ってやつですね。

後半は，まず最初の向かい風がもうアレもんです。ちょうどもどかしいようにアーサーが進みづらくなってますね。あまり怒らないように。神経に触って平常心を崩すのがやつらの目的なのです。ということでカマイタチなんだけどパターンはなし。じっくりといきましょう。運が悪いと上下２段で攻めてきて，ジャンプでもかわせないときがあります。しかたがないからしばらく戻ります。さらに困ったことに，カマイタチってのがカマをもったイタチだってのがこれまた強烈な精神攻撃。俺はもうついていけません。

で，がんばってカマイタチの攻撃を超えると花と茎の攻撃。花のほうはまあいいとして，茎の野郎が最悪。チキンなことに距離を取りすぎると再びニョキニョキ現れます。そこそこ近づいて連射バリバリで殺しましょう。ゲロブタは下から殺すこと。横からファイトすると高くジャンプして寄ってきます。よって，進むべきルートは，まず右へ行ってから上に登るようにしましょう。こうすれば宝箱も現れてグッド。金の鎧は極力残しておくように。武器はタイマツが強そうだけど，実は役に立ちません。槍で勝負です。ちなみに，最上段から左に落ちると宝箱が出現します。

そしてかなり強いボス。パターンは特にありません。結局は左に追い詰められてがんばるのがオチ。火の２段攻撃をしてくることもあるので，ジャンプでかわすときは注意しましょう。このときは，垂直ジャンプのほうが安全かもしれません。斜めに撃ってくる火は極力逃げるほうがいいでしょう。脱がされたら下に突っ込んで上撃ちで勝負するのも手です。それほど硬くないので，要領よく倒してしまいましょう。ということでようやく１面クリア。最終面クリアも近いです。

トリッキーに動く真田虫を見切れ

ブタは下から倒すのが定石

2面もやっぱりハマリます

まずはカメが困る。マメに進むしか手はなく，２連攻撃をされたときにはいったん戻って進むぐらいしかありません。下手に跳ねてしまうと，ちょうど敵が現れてくるので，着実に頭の血管が１本ずつ切れていきます。すでにバーチャルリアリティを超越しています。

んで，蟻地獄もこれまた脳死を早めます。覚えればなんでもないのでがんばります。

と，レッドアリーマー初登場。投げてきた骸骨に当たっても痛いから，浮かれていないように。アルゴリズムがいやらしく，正しい倒し方は見つけられませんでした。上でフラフラしていて，たまに反対側に移動します。このとき上攻撃していれば当てられます。

で，突然突っ込んでくるときに粘れば攻撃できるけど，実際はしゃがんで避けるのに必死でしょう。ときたま地上に降りますが，攻撃すると飛んでいきます（すっごくいやらしい)。飛んでいくときに，うまくジャンプをしながら，やや上方にも武器が飛んでいるようにすると攻撃できることを覚えましょう。タイマツがあるとまあまあ倒しやすいかな。とりあえず，この場所で死ぬと最初に戻されちゃうから，さっさと次までシカトを決め込むのも得策。壁が動き出したら前半終了。

後半も慣れないとやっぱり厳しいから，慣れるまでがんばるのがいちばんの攻略でしょう。ダッセーけど。とりあえず宝箱は４つ確認。結構忘れていってしまうので，武器チェンジだとか，いずれくるアレのためとかに備えて覚えておきましょう。で，ここもやっぱりパターンなし。隙をついて進むべし。植物は倒し損なったからといって焦らず，いったん弾を避けてから殺して進むようにしよう。

ボスはパターンがわかれば弱っちい。まず左に飛んでいったあと，右に戻ってくる。このとき上から降ってくる火の玉に注意するだけ。そのあとは水平に左に特攻，すぐに右に特攻。粘って攻撃してもいいが，しゃがんでかわすことに重点を置こう。あとはじわじわと進んでくる。近づきすぎに注意すれば，ただのダダコネにすぎません。あとはこれの繰り返し。速攻でやっつけましょう。

タイマツだとアツいです3面

エレベーターが昇っていきます。敵の武器の落とし方，宝箱の出現がいやらしくて邪魔。剣，タイマツは最悪です。取ってしまうと上まで届かないから，敵が密集してかなりうるさい状態になります。飛んでるやつが落とす岩に気をつければ特に問題なし。あとはヘボい武器を取らされないように注意するだけ。

そして天井が現れたら右にズンズン進みます。漂っている白いやつがいるけどノープロブレム。こんな雑魚相手してないで早く進まないと挟まれて困ります。

次はそこそこハマる後半。石像の舌に乗っていくわけですが，そのままだと食われるだけ。タイミングよく空中床に乗り移るのだけど，「変な飛んでいるもの」がアッツいことをしてくれます。ジャンプする前に，着地が困らないように弾を撃っておきましょう。移動中に出現する宝箱が邪魔です。基本的には無視するように。こんなところでオヤジと戯れたくはありません。途中，とてもわかりにくいけど舌と岩に挟まれる場所があります。背景と思っていた岩に実は当たり判定があるのです。エッシャー顔負けの騙し絵。こいつは１本取られました，って感じです。ということでここではしゃがみます。最後の舌はさっさと右に進まないと引っ込んでしまうので，速攻で抜けましょう。

いよいよボスだけどこいつもイヤな野郎。台風っぽいけど小さいです。目があるってのはシャレだということにして台風ってこ

ふよふよ動くレッドアリーマー。もーイヤ

君はボスまでたどりつけるか？

とにしときましょう。この台風野郎，しばらくはアーサーの周りを周回します。その間に倒せればベスト。しばらくすると本体が回転します。こうなったら無敵状態なので相手をしないで逃げるべし。回転をやめるとフラフラと気紛れに攻めてきます。軌道がわからないから運よく逃れるようにするしかありません。だいたいこんな感じでさっさと倒すのが肝心です。

臭そうです4面

なんか生臭そうな面です。まずイヤなのがカマ男です。下手なところで殺したりすると，ヤバイ武器を残したりするので安全なところで殺すように心掛けます。

そしてこの4面でさらにイヤなのはミミズみたいな内臓です。このミミズはアーサーのほうに寄ってくるのでそこそこ距離を取って安全に殺しましょう。場所を正確に覚えて速攻で倒せば問題なし。しかし，この面で円盤をもっていると結構ハマります。どうハマるかは自分で体験すると身にしみます。天井に当たると脱がされますので，ジャンプするときは注意しましょう。

ところ変わって凍った滑り台です。ここでは珍しく円盤がとても有効。タイマツもまあまあいけます。イモムシは現れる場所が決まっているので，出現場所で粘ったりしないこと。でかい手は捕まると握られます。ちょっと痛い。普段はハナクソを飛ばしてくるので，ばっちいから避けましょう。なお，手には結構当たり判定があり，攻撃が当たらなそうなところでも攻撃できます。また，この付近は宝箱が多いです。ジャンプしたり，うろついたり，適当に引き返すと現れるので，今後のためにもぜひナイフに変えておきましょう。

そのあとはしばらく下に降りていきます。でかいイモムシが斜めに弾を吐くのがかなりいやらしい。タイミングを計ること。

そしてボス，でかい緑虫。小さい虫は登ってきてアーサーに向かって歩いてきます。強力なのは茶色の真田虫。軌道はランダムで，運よく逃げるのが攻略法です。

タイマツならば，軌道に心臓を合わせておけばまあまあ倒しやすいと思われます。円盤は困ったことに全部ジャンプして下攻撃でなければ心臓を潰せません。ただし小さい虫が殺しやすいのでちょっと嬉しいかも。槍は左右それぞれ2つ目の心臓を横から潰せます。ナイフならば左右2つずつすべて横から潰せます。端ギリギリのところで連射あるのみ。小さい虫の処理はしっかりするように。剣，斧の人はもう死んだほうが早いと思われます。

最終的に中央の心臓はジャンプして下撃ちするのですが，これはもうアドリブで避けるしかないです。無理せずに小さい虫は積極的に処理。ちなみに真田虫は端のほうまではあまりきません。

小ボス愚連隊5面

まず前半ですがここまできた人にはたいしたことはないでしょう。レッドアリーマーは1匹ずつ倒し，ブタは全部下から殺すように。最後のブタはタイミングを見て突っ込みましょう。

で，5面の後半がアッツい。まずハシゴを上がるところですが，安全なタイミングが必ずあるのでパターンを作って上がろう。上がるとゴリラが登場しますが，恐れずに近くに寄ります。ジャンプして顔を撃ちまくれば速攻で倒せますし，火炎放射攻撃は近場でしゃがめば大丈夫。

そのあとの盾顔野郎は，ハシゴを上がったあと左右ほぼ同時に火を吐くので注意。これを中央で左右どちらかにジャンプしてかわしたあとは，顔の下に潜って片方を速攻で倒します。あとの1体はアドリブで倒せるでしょう。ちなみに後半スタート地点に宝箱1個，ここの手前に1個あるので金の鎧を保険として出しておくと楽になると思います。

そのあとはゴリラ2匹。右側に近づいて速攻で倒します。左は無視してかまいません。そのあと，段差のところには小さい台風が3匹いるので，1匹ずつ確実に倒していくハメになります。逃さず確実に数を減らすように攻撃を集中しよう。逃すと追っかけてきますので，その近辺で必ず仕留めないとどうしようもありません。ここの斜面の連続にも宝箱が2つあるので鎧を残すようにしましょう。

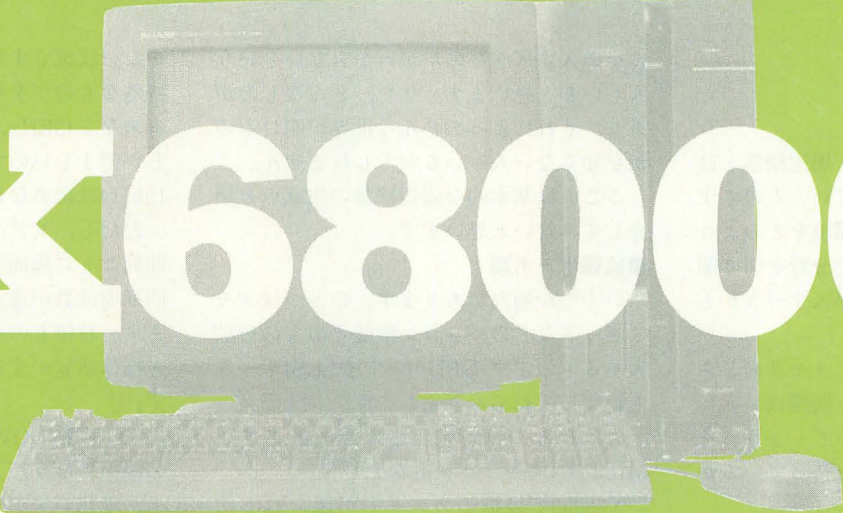
いよいよボス，蠅野郎。まず小蠅が密集すると本体が現れ，ケツから弾を3回垂れます。これがかなり避けにくいのでなるべく画面外で撃たせるように距離を離してスクロールさせます。と，すぐに分離して攻めてきます。距離を十分取っておけばその間で再び合体し始めます。急いで左右どちらか，スペースの広いほうへ逃げておきましょう。以降，これの繰り返し。この間にうまく攻撃を加えます。ボスが合体してから，ほんのわずかな間に攻撃が当たるように弾を残しておきましょう。運が悪いとハメを食らいます。小蠅で移動中にも当然のように当たり判定があるので距離を十分取っておくのが解決策です。

面倒臭いだけの2周目

ということでなんの脈絡もなく2周目に突入してしまいます。まいったね。

2周目では新しい武器，サイコキャノンを手に入れられます。これがないとクリアできません。最強兵器なので1周目よりも楽に最終面に到達できるでしょう。ということでラストボス。

きっと偉いんだろうけど，怠慢な野郎です。ジャバザハットほど太ってないけどやっぱり歩けません。裸なので照れちゃってるんでしょうね，きっと。攻撃はレーザーと足で踏みつけです。レーザーに注意して，足の上に乗って顔を攻撃すればかなりあっけなく死んでくれます。突っ込みが肝心でしょう。

終わり

攻略になってないです。ごめんなさい。結局パターンが一切作れないのも難しさのひとつなんでしょうな，きっと。ここまで脳味噌を直撃するゲームには久しぶりに出合った気がします（おかげでいつもと違う文体だし）。面白かったけど。

んで，さっそうと次回作へのリクエストをしてみます。俺としてはカプコンさんに「戦場の狼2」とか「1943」とか「サイドアームス」とか「エグゼド」を出してほしいね。クイズモノもいいよな。

［特集］

X68000と仲間たち

X68000

この春はついに新機種が発表されずじまいとなり，ちょっと寂しいハードウェア状況となってきました。
X68000はすでに整った環境を持ってはいるものの，世間の流れを見るともっともっとパワーをと望んでいる人も多いようです。
たとえ本体は変わらなくても，周辺機器によってパソコンをパワーアップさせていくことは十分に可能です。
見落としてはいけないのは，ハードの力が発揮されるのはソフトウェアの対応があってこそです。
ドローツールの登場によりレーザープリンタは無意味ではなくなりました。音楽ツールがあってこそMIDI楽器も生きてきます。
新しい周辺機器は常に開発されていますし，機能/性能ともどんどん向上していきます。現在，それを駆動するよいソフトウェアがないためにX68000では性能を生かし切れないものがあるのも事実です。よいハードが増えれば増えるほどよいソフトの充実が望まれるところです。

CONTENTS

周辺機器事情'94……中野修一
これが噂の040turbo……紀尾井誠
サブCPUシステムPOLYPHON-24……瀧　康史
メガディスプレイへの道……瀧　康史
当代ハードディスク事情……伊澁見あきら
ちょっと大きめの常用HD……友永健明
MOドライブの一般的使い方……中野修一
汎用MOユニットを接続する……紀尾井誠
3.5インチ光磁気ディスクドライブの活用……丹　明彦
満開式磁帯駆動装置壱號……友永健明
さらに進化したサイバー絵筆……川原由唯
謎のPPIボードただいま見参……瀧　康史
無停電電源装置BX3……紀尾井誠

パワーアップアイテムあれこれ

周辺機器事情'94

Nakano Shuichi 中野 修一

周辺機器といってもいろいろあります。ここでは広く周辺機器一般について，どんなものがあるのか，X68000シリーズで使うにはどのようなものを選ぶべきなのか，どんなことに注意すべきなのかを中心に解説してみましょう。

それは可能性

あらためていうまでもなく周辺機器とはパソコンをパワーアップしていくものです。自分が使っているパソコン環境をどのように変えていくか，どういった分野を切り開きたいのかによってなにを導入すべきかということは変わってきます。

たとえばハードディスクとメモリのどちらを先に導入すべきかという問題は，人によって返ってくる答えが違うでしょう。それぞれに違う目的と価値観を持っているからです。

パソコンは使い手とともに進化していきます。周辺機器はそれをいっそう大きく前進させます。拡張による絶えざるパワーアップはパソコンユーザーの共通した願望だといえるでしょう。

拡張は，突き詰めれば際限のない分野です。ともすれば拡張自体を目的にしているような人も見受けられます。しかし大事なのは，あくまでもそれによって達成される世界が広がっていくということです。別に「使いこなす」「極める」といったレベルになる必要はありませんが，しっかりした目的を持っていれば，それは自然に流れ着く場所でもあります。

もちろん，たいそうな目標がなくても，周辺機器は手軽にパソコンの利用範囲を広げてくれます。しかし，単に周辺機器を買ってきてつなげばいいかというと，たいていはそれだけではいけません。OSやソフトウェアの設定についても考える必要があります。

それはとりもなおさず，「環境を考える」ということになります。

周辺機器あれこれ

目的ははっきりしていても，具体的になにを導入すべきなのかがわからないという人もいると思います。また，どんなものがあり，それによってどんな世界が開けるのかを知らない人もいるかもしれません。

ここでは基本的な周辺機器の注意点を紹介してみたいと思います。

●拡張ボード類

いくつか傾向があります。ひとつはメモリボードなどのパソコン機能そのものを拡大するものです。CPUボードやSCSIボードも範疇に入れておきましょう。

もうひとつは特定用途のボードです。MIDIやスキャナボードなどです。

X68000では拡張スロットが限られるため，どうしても特定用途のボードを導入することにためらいが出てしまいます。特に非SCSI機ではしっかりした目的がないとなかなか手が出せません。

優先順位はメモリ，SCSIとなるでしょうから，スキャナはRS-232CかSCSI，ビデオボードは外部ユニットに，MIDIはRS-232Cに，FAXボードはモデムで，というふうに代替すべきでしょう。

また，最近になってようやくアクセラレータや同種のCPUボードが登場しそうな気配になりました。これまでもCONCERTO X68やV70ボード，POLYPHONなどといったCPUボードがあったのですが，既存の環境をそのまま高速にできるものはありませんでした（REDZONEとかはあったけど）。

本体そのもののパワーアップができ，さらにHARPや040turboはスロットを占有しないので拡張ボードとしてはもっとも望まれているタイプのものだといえるでしょう。

●CRT

X68000初期型当時の古いディスプレイを使い続けている場合，そろそろヘタってきている人も多いかと思われます。

コストまで考えると現在は適当な代替品がない状況ですので（42ページ参照），多少特殊なものに手を出すことになるでしょう。本来なら15kHz，24kHz，31kHzが映ることが望ましいのですが，高解像度機種では15kHzは諦めなければなりません。

ただし，リアルタイムゲームは垂直帰線期間ごとに画面を描き換えますので，垂直同期周波数が違うと動作速度が変わってきます。移植もののゲームでは15kHzでないと操作感覚が変わってしまうことがあります。

SX-WINDOWなどを使うなら，大きめの画面のほうがよいでしょう。20インチ以上が理想ですが，あまりにも高価なので手ごろ感のある17インチに落ち着きます。

モードによって画面の大きさなどを変えることの多いX68000ではナナオF557が便利でしょう。モードごとに設定状態を記憶させておけば自動的に調整してくれます。ただし，こういったディスプレイは15kHzに対応していないのが難点です。

価格と周波数対応範囲などから飯山電機のHelloIIシリーズもよいでしょう。これも15kHzには対応できませんが。

画質はさほどではありませんが，エプソンCR-7000は中残光ディスプレイであること，15kHzに対応していることなどからX68000に適したディスプレイであるといえます。特にSX-WINDOWでの使用には適しています。ただし，現在では入手が非常に難しいと思われます。

輸入もので，DOS/Vマシン用の信じられないくらい安いモニタがあったりしますが，あっという間に壊れたり，使うたびに輝度が落ちていったりすることもあるらしいので注意しましょう。

●ハードディスク

SCSIタイプしかすでに入手できなくなっています。一世を風靡した240Mバイトタイプが格安状態。384Mバイトが手ごろな値段というところでしょうか。詳しくは伊遊

見氏の記事を参照してください。

●光磁気ディスク（MO）

現在，「光磁気ディスク」といえば3.5インチタイプを意味します。

用途はハードディスクバックアップ用とかデータストレージ用，データ輸送用などいろいろありますが，すでにかなりフロッピーディスクに近い感覚で扱われるメディアになってきています。

特にX68000ではかなり普及してきたといってよいでしょう。昨年のコミケではついにMO版の同人ソフトまで現れていましたし（といっても福嶋君のとこだが）。

●CD-ROM

MOに比べると普及率は極端に低くなります。計測技研からCD-ROMドライバが発売されていますから，それとドライブさえ買ってくればすぐに使えます。問題はもちろんアプリケーションソフトのほうなのですが……。

最近は3倍速タイプが登場し，2倍速タイプはかなり安くなってきました。こういうものは基本的に速いほうがいいに決まっています。そのほか，最近のものならPhoto CDマルチセッション対応などはごく普通に見られるようになりましたし，特に機能的な心配はないでしょう。

人によっては音楽データをデジタルデータとして読み込める最近のドライブを選んだほうがいいかもしれません（ソニードライブ，東芝ドライブ使用品）。

また，MPEGボードと併用する場合はMPEGが読み込めるドライブでなければなりません。

●MIDI機器

X68000では旧型のローランドSC-55が主流です。

SC-55mkIIとの互換性は完全ではありません。先日ついにCM-300が生産完了となったため，旧GS規格の音源ユニットの入手はますます困難になってきました。

今後，SC-55mkIIが主流になっていくのか，一気に新機種のSC-88あたりにいくのか，それともGM音源レベルで落ち着くのか，まったく不明です。

一般に楽器メーカーは互換性に対する認識が薄く，後継機でも同じツールが使えなかったり，パラメータの扱いが変わっていたりということがよくあります。

純粋に音楽制作に用い，DTM音源としてデータの共有を考えないならば，これはもうなんでもかまいません。各自の好みで決めてください。

●ビデオ入出力装置

カラーイメージユニット（CZ-6VT1）とビデオボード，そして先頃発売されたビデオ入力ユニット（CZ-6VS1）の3種があります。

大雑把にいってしまえば，カラーイメージユニットのビデオ出力部分を取り出して高性能化したのがビデオボード，入力部分だけを取り出して高性能化したのがビデオ入力ユニットといえるでしょう。なお，入出力がともにできるのはカラーイメージユニットだけです。

パーソナルアニメーション制作などではビデオボードが必須ですし，ビデオからの取り込み画像はビデオ入力ユニットが綺麗です。

ただし，ビデオボードはそのまま使うとスロットを2つつぶしてしまいます。本体の拡張スロットから供給しているのは電源だけですから，できれば改造してユニットタイプにしてから使いたいものです。そのほうがノイズにも強くなると思われます。

ビデオ入力ユニットも接続にSCSIを使用するためEXPERTII以前の機種ではI/Oスロットを使用しなければなりません。広告とは裏腹に，転送のためにSCSIバスを占有され，表示はCPUが行うのでパソコン本体（CPU）の負担は増えています。

●プリンタ

シャープ48ドットカラーがいちばん普及率が高いのですが，最近はキヤノンBJシリーズがずいぶん普及してきました。実用面からいくと，BJ-10/15シリーズはいくら安くてもおすすめできません。BJ-220JCあたりが手ごろです（いずれも，アウトラインフォントつき機種はおそらく無駄になるでしょう）。

カラープリンタを選ぶならいまはBJC-600Jしかないでしょう。

最近はレーザープリンタが手ごろになってきましたので，思い切ってそちらを選ぶ手もあります。X68000で使う場合PostScriptはほとんど意味がないので，レーザープリンタも極端に高いものではありません。低消費電力型のキヤノンLBP-A404GIIで定価148,000円と，パーソナルユース機では実売10万円を切るようなものもあります（秋葉原価格）。SX-WINDOWでの使用を考えるとキヤノンLIPS 3かエプソンESC/Page対応機がよいでしょう。

そろそろ高嶺の花だったレーザープリンタも旬になってきたようです。

●モデム

日進月歩で商品サイクルが非常に短かく，具体的におすすめの機種を挙げることは困難です。機器の性格上，よほど特殊なものを除き，つながらないものはないと思ってもかまいません。

性能的に問題があっても使用に耐えないというほどではありませんし，有名メーカーのものが優れているというわけでもありません。定評があるモデルでもモデルチェンジで評価がまったく変わってしまうものもあります。急に大きさや形が変わったりしたら前の機種とは別ものと考えたほうがいいでしょう。

ホストとの相性などの細かい問題もあるので，まずはなんでもいいから安いのを買っておき，あとはそれを使って情報収集するというのが確実です。

通信速度については高速なもののほうがいい，というのは当たり前ですが，値段との兼ね合いで決めるべきでしょう。また，いくら28800bpsのモデムが現れたからといっても，ホスト側が対応していなければ無意味なのはいうまでもありません。

そのほか，X68000の市販ソフトではFAXモデムを扱うものはありません。しかしX68000でFAXを扱えるフリーウェアも発表されていますから，それを入手できればFAX機能つきのものを選んでも無駄にはなりません。というより，最近のモデムはもれなくFAX機能がついているでしょうから，ぜひとも積極的に活用したいものです。

●マウス

X68000用としてスピタルからSYGNAS Xというマウスが発売されています。カウント可変タイプでさまざまな機能が搭載されています。ややボタンの遊びが大きいようなので，編集部にあるものは少し改造が施されました。さすがに標準マウスよりは使いやすいようです。

そのほか，マウス変換アダプタ（東京システムリサーチ，満開製作所など）を使えばPC-9801用のバスマウスが使用できます。

定評のあるのはマイクロソフトマウス（特に旧型）などです。種類が多いので各自好みのものをみつけてください。

先日，八重垣氏おすすめのエレコムGRAIOを試用してみました。確かにボールの取り付け位置などは正しいのですが（マウス前端にある），正面方向が斜めにずらして設定してあり，かなり操作感覚が異なります。この機種は癖が強く慣れが必要なようです。

●ジョイスティック

あまりおすすめのものはありません。
技術のある人は自作しましょう。

[特集] X68000と仲間たち

見果てぬ夢を目指して

これが噂の040turbo

Kioi Makoto 紀尾井 誠

X68030に68040を搭載してしまおうという計画から生まれたパワーアップボードです。基板を取り出し，シールド板をはずして，CPUを抜いて，冷却ファンをつけて，といろいろ作業は必要ですが，その甲斐あって現在のX68000シリーズでは最高峰の速度を叩き出します。

X1時代から恒例だった春の新製品発表もなく，ハードウェアの話題が乏しい今日この頃。

MacintoshではPowerPC機が話題になり，CPUパワーに関する議論も活発になってきています。振り返ってX68000の状況を見るとやはり寂しいものがありますね。

X68000ユーザーなら誰しもシャープさんに新機種を，

「早く出してくださいね」

というのを河内弁で訴えたい気持ちでいることと思います。

石上氏のアクセラレータはまだ苦戦中，一時話題になったジャストのHARPはいまだ姿を見せず，といった状況のなか，順調な仕上がりを見せているのが今回紹介する「040turbo」です。

68040とはなにか?

68040とはX68030で使用されている68030 (68EC030) の上位にあたるCPUです。後継となる最新作，68060の出荷がまだ行われていませんので，68Kファミリーでは現状で最高速のCPUとなります。

モトローラのCPUは68000で68000個のトランジスタ，68020で20万個のトランジスタ，68030で30万個のトランジスタ……とわかりやすい型番がついていたのですが，68040では一挙に120万個にまで集積度が上がっています。内部は大幅にワイヤーロジック化され，基本命令はすべて1クロックで動作します。「基本」でない命令は数クロックを要しますが，実際のプログラムを平均すると1.3クロックで各命令を処理するといいます。

68000の命令処理クロック数が最低4クロックで，単純なMOVE命令などでもアドレッシングモードによってはすぐに10クロック以上になっていたことを考えると，桁外れの性能になることが予想できるでしょう。68000の4倍程度の性能を持つ68030でさえ，最短命令は2クロック（同期バス時），たいていの命令で4～8クロックかかっていました。20クロックくらいの命令もたくさんあります。68000に比べれば大幅に改善されているものの，乗除算などには目を覆うような数値も残っています。それが平均1.3クロックになるというのです。

が，こういうものにはたいてい「キャッシュがヒットした場合」という条件がつきます。もちろんメモリもノーウェイトの状態です。キャッシュがはずれると，その補塡にかえって時間がかかります。

さらにキャッシュがOFFの場合はパイプラインがスムーズに流れませんから，それほどの性能向上は期待できません。

高性能CPUでは，いかにしてメモリのアクセスを減らすかというのが大きなテーマになっています。当然，キャッシュの挙動が全体のスループットを決定する大きな要因になります。ここで68040のキャッシュ制御を見てみましょう。

キャッシュは命令キャッシュとデータキャッシュに分かれており，命令の取り込みとデータのアクセスを並行して行うことができます。それぞれ4Kバイトの容量を持っています。

ちなみに，1ライン16バイトで4Kバイトの命令キャッシュを持ったCPUのキャッシュヒット率は約90%といわれています。256バイト容量のときには75%程度とされていますが，最近のX68000のプログラムではループ展開などは当たり前ですから，これほどには当たらないかもしれません。しかし，X68030ユーザーの方ならキャッシュの威力というものをよくご存じでしょう。たった256バイトずつのキャッシュでも相当の効果を上げています。キャッシュサイズが大幅に増えたことにより，ある程度展開されたプログラムでもキャッシュの恩恵に与れそうです。これでも効かないのは，SLASHの画面消去ルーチンくらいのものでしょう。

●コピーバックキャッシュ

さて，68030でもキャッシュによりメモリからのデータ読み込み頻度は大きく減らすことができました。しかしデータ書き込み回数はまったく変わっていません。68040では，書き込み時にもキャッシュが効くコピーバックモードが追加されています。

従来，メモリへのアクセスが発生した場合はそのままメモリ内容を更新していましたが(ライトスルー)，キャッシュの内容だけを更新して，必要になるまではキャッシュの中身をメモリに書き写さないようにすれば，メモリのアクセス回数をかなり減らすことができます。

このような方式のキャッシュ制御を一般にコピーバックといいます。

OSレベルで見れば，キャッシュ関係の命令は変更されていますのでそれらを使ったプログラムでは変更が必要になります。ただし，IOCSコールを用いてキャッシュコントロールしているプログラムならば040turbo側が対応しますので問題はありません。

●浮動小数点演算ユニット内蔵

浮動小数点演算ユニットを内蔵しています。関数などの処理はソフトウェアでやらなければならないのですが，基本演算性能は68882の20倍だといわれています。

*　　*　　*

メモリアクセスで，68030ではバーストモードというものがありました。キャッシュミスが起こったときのキャッシュ再充塡を通常のメモリアクセスよりも高速に行うためのものでした。

本来68040はバーストモードを基本として動作します。さらに，その高速なメモリアクセスを一般の命令レベルで開放されていたのですが，X68030のハードウェアではバーストモードがサポートされていないの

でその恩恵には与れないようです。非常においしい機能だけに残念です。

040turboとは

040turboとはX68030のCPUの代わりに68040を載せたドータボードを装着しようというものです。従来のプログラムが完全に動作するわけではありませんし，あらゆるプログラムが必ずしも高速化されるというものでもありません。アクセラレータとしての68040ではなく，68040自体を動作させることを目的としたボードなのです。

040turboに関する書籍はソフトバンクから発売されますが，ボード自体は計測技研から発売されることになります。

そもそもの発端は，作者の大塚氏（BEEPs）がシャープからの6年目の回答としてのX68030に不満を感じて行動を起こしたところから始まっています。氏はかつての「X68000あなたの知らない世界」での常連でしたので覚えている人もいるかもしれませんね。

X68030発表時点で見ても，明らかに68030（25MHz）のハイエンドマシンというのは1世代以上前の仕様でした。

「なければ作る」

というのがX68000ユーザーの基本スタンスであり，そして実際に作れるユーザーがいるということが現在のX68000を支えています。その観点からすれば，このような動きがあっても不思議ではありません。POLYPHONやDSPボードなども同様な流れから出てきたものだと思われます。

その一方で，進化を続けるフリーソフトウェアの前にメーカー純正のツールはどんどん駆逐されていきました。

そして，暗示的なことに040turboは「フリーハードウェア」なのです。

040turboを装着する

040turboはX68030のCPU部分の上に装着します。スペースの関係で5インチタイプでしか使用できません。3.5インチタイプではまったく余剰スペースはありませんから，もしやるとしても完全に基板をバラバラにしたままでなければ装着はできないでしょう。

作業は本体をバラすところから始まります。事前にある程度の工具を揃えておいたほうがいいでしょう。

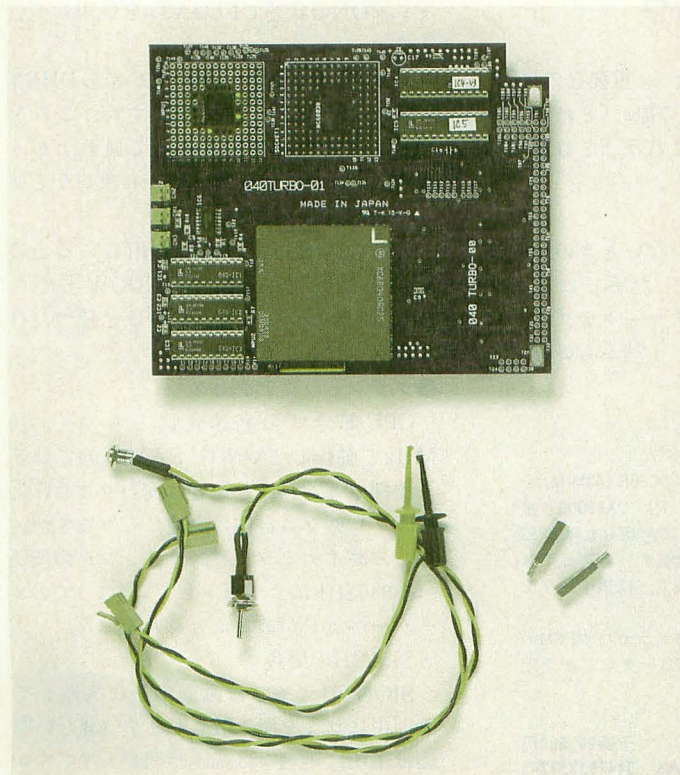
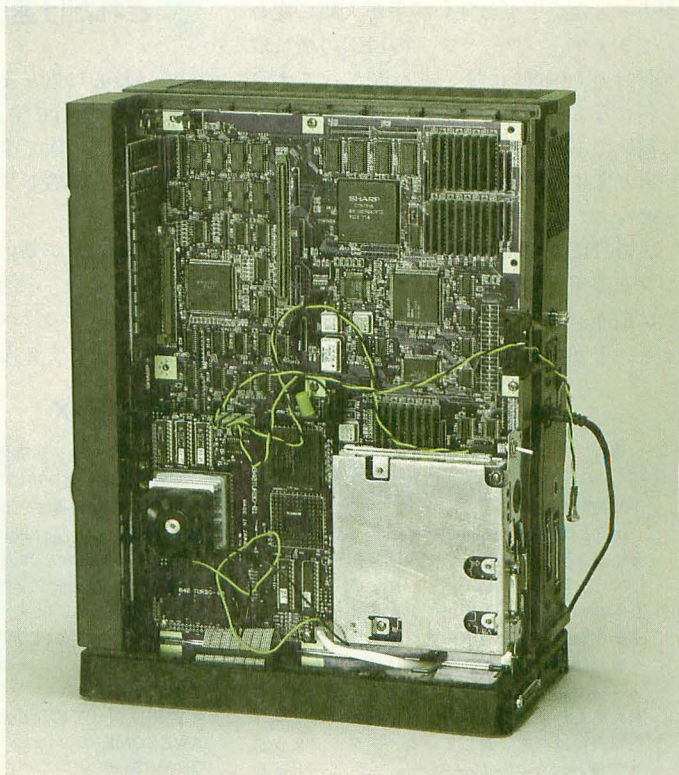
メインボードを完全に取り出すところまでやらなくてはならないので，これまで一度もバラしたことのない人にはちょっと難しいかもしれません。この部分だけでもマニュアルには実にきめ濃やかな解説がなされています。

とりあえずバラします。CPUを抜きます。クロックを取り出すICクリップをつけ，CPUソケットに040turboを載せてみます。増設メモリや拡張スロットはなくても立ち上がりますので，RGBユニットだけをちょこんと載せて起動してみましょう。

電源を投入してすぐにOFF。POWERランプがきちんと点滅すればよし，消えなかったり，すぐに消えた場合は接続がうまくいってませんので背面の主電源を切ります。

私が最初に試したときはうまく接続できていませんでした。基板を押し込んで再度電源を投入，まだだめです。基板をはずして，クロックを取るクリップをつけ直し，慎重に基板を載せます。再々度挑戦。どうやら，うまくつながったようです。ボードが平均して差さっているかどうかに気をつけるとよいようです。

ここで取り付けたICクリップはCPUに分周前の50MHzクロックを与えるためのものです。68040は2相クロックで動作しますので，25MHzのクロック以外に外部クロックに同期した倍速のプロセッサクロックが必要になってくるのです。

マニュアルではクリップではなくハンダ付けしたほうがよい，となっていますが，まったくそのとおりです。今回は借り物ですのでICクリップで信号をとってきていますが，基板の隙間と拡張スロットの端ぎりぎりの場所というかなり微妙な位置に収めなければならないため，拡張スロットの取り付けはみあわせました（基板のバージ

040turboの基板と付属品。製品の予価は98,000円（税別）

基板を装着したところ

ョンで違うこともありうる？)。

しばらく動作させてみるとCPUがかなり熱を持っていることがわかります。短時間でも68030とは大違いです。ここで使ったものにはあらかじめ放熱器と冷却ファンが取り付けてありましたが，ファンを回す電源が確保できないのでしばらくうちわで扇ぎつつ動作をチェックしていきました。このあたりの冷却措置は基本的にユーザー任せです。

システムディスクを作り，動作させてみるとだいたい動きそうです。さすがにこのままでは細かいチェックができないので（冷却なしでの耐久テストはできるが，ちょっとやりたくない），あまり望ましくないのですがメインボードから電源を盗んでファンを回してみます。5Vでは弱いかと思われましたが，意外と十分に冷却効果があるものです。

冷却は大丈夫そうなので残りの部分を組み上げてみましょう。

040turboは基板から引き抜いた68030をボード上に載せることで，68030と68040をスイッチで切り替えることができるように設計されています。スイッチは本体背面のアース端子部分をはずして取り付けます。

オプション部品として，現在起動中のCPUが68040であることを示すLEDがあります。もちろん本体にはそんなものを取り付けるスペースはありません。しかしこれがあるとないとでは使い勝手が大幅に変わってきます。パワーランプのLEDの横に取り付けると040時に緑と赤が混ざってオレンジになる……かな思ったのですが，実際に置いてみたところ，ちょっと苦しいものがあります。角度によってはまったく見えません。

まあ，HD内蔵機でなければHD BUSYランプを流用する手もありますし，いざとなれば前面パネルに穴を開けるのもいいでしょう。どうするかはユーザー任せというところです。

まずは68030モードで起動してハードディスクのシステムを040兼用に作成します。基本的にはデバイスドライバとして，

040syspat.x

を登録します。

このパッチドライバは68040をキャッシュONの状態で立ち上げるために必要になります。このドライバがなくても起動はできるようですが，キャッシュは動作しないので素直に030モードで使うほうがよいかもしれません。

040時にFLOAT4は使用できませんので，

DEVICE＝FLOAT040.X

DEVICE＝FLOAT4.X

のように順に記述することでCPU別に使い分けることができます。

そのほかの問題があるソフトを共有するためにはダイナミックパッチャを使用します。これはソフトをメモリに読み込む時点で登録されたファイル名に応じたパッチデータを加えてから起動するものです。コンフィグファイルにパッチ情報を追加していけば複数のソフトに対応できます。デフォルトではFSX用のパッチ情報などが記述されています。

そのほか，すでにいくつかのソフトウェアがこのボードに対応しており，GCCの040版なども収録されています。

どれだけ速いか？

では核心に迫りましょう。重要なのは「どれだけ速いか」です。この際，「どれだけ走るか」は二の次です。これは「どれだけ動かすか」と同義ですから。

●体感

キャッシュが効いていないときの速度はX68030をちょっと重くした感じになります。こういうのはベンチマークテストではなかなか表れませんが，1割くらい遅くした感じでしょうか？

次に，キャッシュをライトスルーに入れます。これでひと心地ついた感じで動き出します。通常のオペレーションでは劇的な速さというのは味わえませんが，X68030でやっても重かったようなことをやらせると初めて真価を発揮します。

さらに，多くのソフトで問題はあるのですが，コピーバックモードを入れてみます。一段と高速化されるはずですが，体感速度にはあまり表れてきません。

●演算速度

ベンチマークテストとしてOh!Xで標準的に扱われているDHRYSTONE, WHETSTONE, STANFORDの3種類をXC ver.2でコンパイルしたものを使ってみました。実行ファイルは中森章氏がV70ボード評価のときに使っていたものと同じです。XC版を使うのはこれまでのデータがそうだったからという意味もありますが，GCCによるものよりメモリアクセスが多く，より現実的な数値が期待できるからです。

表1に結果をまとめます。それでも体感速度とのあいだには差があります。キャッシュオフ時の動作などを見る限り，STANFORDがもっとも自然な結果を示しているようです。X68000(10MHz)＋FLOAT2 ver.2.0を1としたSTANFORDの結果をグラフにしてみました（図1）。

対照機種となるX68030(25MHz)はキャッシュONの状態でFLOAT4を使用しています。

コピーバックモードでXCでのDHRYSTONE値は12345.7程度ですが，これをGCCにするだけで倍近くまで跳ね上がります。メモリアクセスがいかに重いかということがわかります。

X68030を使っていても利用者に暴走の危惧を抱かせるSX-WINDOW上でのJPEG画像のロードも，10秒ほど待つだけで表示されるようになります。

●グラフィックアクセス

CPU部分での高速化はともかく，10MHzで動作しているG-RAM部分ではどの機種でもそれほど速度は変わりません。描画プログラム自体は高速になりますが，むしろダイナミックバスサイジングの関係（68040自体はサポートしていない）でのオーバーヘッドがあるかもしれません。

・SLASHの場合

SION4のデモでは画面消去に失敗して画面中にゴミを撒き散らします（詳しい原因は不明）。処理中は描画時間が大半ですからそれほど速くはなりません。表示される

DSPボードAWESOME-X

現在，7月発売を目指してX68000用パワーアップボードとして開発されているのがDSPボードAWESOME-Xだ。

DSPとはDigital Signal Processor，デジタル信号処理専用のプロセッサだ。演算処理に特化したCPUと考えてもよいかもしれない。プログラムを書いておき，入力ポートから流れ込んだデータを加工して出力ポートへ送る，というのが一般的な使用形態だ。たとえばリアルタイムのフーリエ変換とか，JPEG圧縮とか，その他，行列の絡んだような演算を大量に行わせるといった場合には非常に適している。スターフォックスで使われたスーパーFXチップやバーチャレーシングのチップも実体はDSPだ。

AWESOME-XはDSPにTMS320C26B(40MHz)を使い，共有RAM領域（4Kバイト）でX68000と接続されている。128,000bpsまで対応したRS-232C×2ポートほか，赤外線通信インタフェイス，光端子によるオーディオ入出力，拡張インタフェイスを備える。

ソフトウェアはFLOAT互換ドライバやPCMドライバ，JPEGエンコーダ/デコーダなどが予定されている。

詳細は追って紹介したい。

AWESOME-X　予価90,000円
GRAVIS　FAX 044(813)7243

数値はほとんどが1か2です。これは1/100秒タイマによる画面表示間隔を表しています。ですから，だいたい秒間60コマ以上の描き換えが行われていることになります。ただし，最悪時には瞬間的にF（15）が発生します。

ALLTESTAではキーのチェックにうまく引っかからないためか，1回角度を変えるあいだに何回も描き直しているようです。かなりちらつきが発生します。概ね速くなったとはいえるでしょう。

・SION2の場合

SION2（X68030対応版）で試したところ，X68030キャッシュON時に「たまに画面消去に失敗」していたものが，040turboでは「たまに画面消去に成功」する，という感じになります。画面は線で埋め尽くされますが，ウェイトなしでもほとんど重くなるところはありません。

そのほか，赤ずきん関係では60人踊るとかなり重くなりますし，往復ビンタは295発くらいになります。

どれだけ走るか?

二の次の出番です。

試用してみたところ，付属のシステムパッチドライバとダイナミックパッチャのおかげか，予想以上にソフトが動作します。

こういったものが威力を発揮するのはプログラム開発現場でのコンパイル，アセンブル作業か，SX-WINDOW環境なのですが，そういった局面で使うツールは比較的おとなしめですから，ほとんど問題なく動作しています。

なんとなく「危ないかな？」と思うようなものは，やはり引っかかります。68030のキャッシュONで動作していないものは当然引っかかります。X68030が出てきて1年たつもののまだまだ自己書き換えを行っているアプリケーションも多々あります。68030では問題なくても68040では顕在化する問題点もあるでしょう。自己書き換えの代表例であるLZXはかなり支障があります（プログラムによる）。Z-MUSICでは演奏処理には問題はありません。コンパイル時などの細かいモード設定によっては支障が出る可能性もあります。

キャッシュのモードはコピーバックがおいしいのですが，通常使用ではライトスルーモードを選択しておいたほうが無難です。テストユーザーによる対応が進んでいるようですから，そのうちコピーバックでも安心して使用できるようになるのでしょうか。しかし基本的に，こういうものは「走る」ものではなく，「走らせる」ものだと思っておいたほうがいいでしょう。

物理的な意味で「どれだけ走るか」を見ても，ほとんど問題はないようです。小なりといえど冷却ファンのおかげか，丸1日以上連続動作させても安定していますし，チップの過熱もたいしたものではありません。むしろ冷却していない68030のほうが発熱は大きいようです。

最後に

という原稿を現在X68030＋040turboのSX-WINDOW上で書いているわけです。立ち上がる環境を設定することができれば，それほど大変なことではありませんでした。コピーバックモードではないので極端に速くなったという感じはありませんが，かなり自然な動作になります。これは編集室で普段使っているSX-WINDOWシステムのデスクトップが重いせいもあります。

PCM8を組み込んでSXZCでウィンドウ上からカラオケデータを次々に再生しながら，RSDRV.SYS + CommunicationSX-68Kで通信するという危なっかしい使用状況に耐えているのですから，かなりがんばっているなというのが正直な感想です。

なお，デフォルト状態のSX-WINDOWならばコピーバックモードでも動作しますが，この状態なら驚くほどの速度でウィンドウが開きます。

Oh!X編集室では，編集作業のかなりの部分がSX-WINDOW上で行われています。040turboでSX-WINDOWがかなり素直に動いているので巷の評判はよいようです。

しかし，現在のX68000のI/Oの上に高速なCPUを載せたとしても，それほどの性能向上は見込めません。ドータボード上にローカルRAMと2次キャッシュを積むなどの手もありますが，I/Oでのオーバーヘッドはいかんともしがたいものがあります。

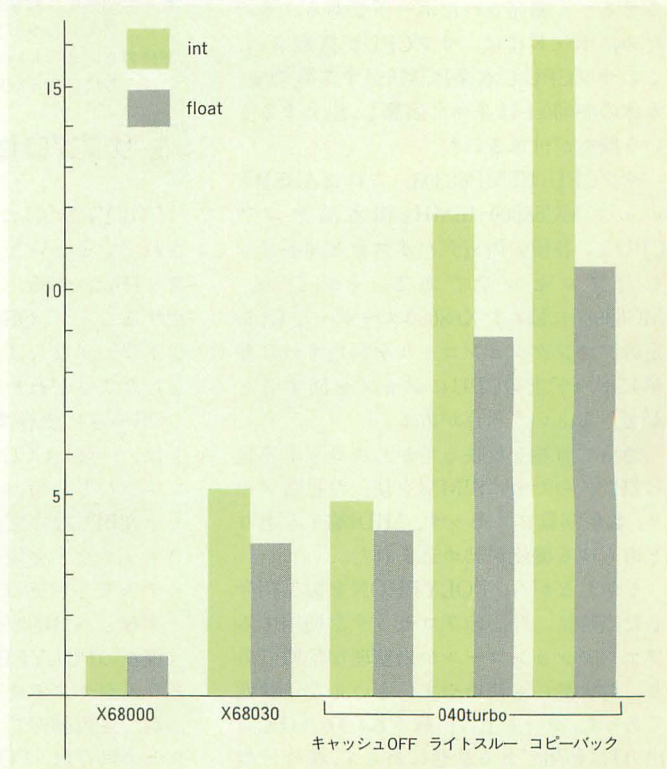
システムのバランスから見るとX68000はちょっとおつむが軽い感じ，X68030はやや頭でっかち，040turboではかなり手足が重い，といった感じでしょうか。

さて，最後になりましたが，040turboというものは，はたして必要なものなのでしょうか？　X68030自体はそれなりにバランスのとれたマシンですし，使用していてそれほど不満はありません。メーカー保証の範囲外のボードですから，本来はこんなことをしたくはないのですが，ハイエンドマシンが2世代前の仕様しかないという状況は打開せねばならないような気もします。結局は「必要なかったらよかったのにね」というところに落ち着いてしまいます。

半ば公然とした本体改造にしても，メーカーの方にはそこに込められた「叫び」を感じとってもらいたいものです。

表1　ベンチマーク結果（単位ms）

	X68030	040turbo		
		OFF	ライトスルー	コピーバック
DHRYSTONE	279.2	188.6	123.2	81.0
WHETSTONE	3.38	2.97	2.01	1.77
STANFORD int	108.1	118.4	48.1	35.2
STANFORD float	672.3	591.7	275.8	232.9

図1　STANFORDベンチマークの性能比

伝説のPOLYPHONは健在か？

サブCPUシステムPOLYPHON-24

Taki Yasushi 瀧 康史

PCM8をサブシステムで実行，各種拡張ボードを1枚に集約……と鳴り物入りで登場したPOLYPHON。現在も地道に進化を続けています。今回は24MHzバージョンと新機軸ソフトウェアPOLYPHONシステムを紹介します。

伝説のPCM8

X680x0ユーザーの間にセンセーションを巻き起こしたあのPCM8は，読者の方々の記憶にもまだ新しいことだろう。

本体無改造でPCMを8チャネル同時に使え，IOCSのADPCMOUTでもポリフォニック*1)する構造は非常にセンセーショナルであり，私も含め，読者のなかにもFLOAT?.Xと同じくらいシステム構築の必需品となっている人がいるに違いない。

しかしながら，PCM8は猛烈なマシンパワーを必要とした。マシンがXVI以上ならば，常駐していても一向にかまわないのだが，10MHz機ではPCMがポリフォニック再生するたびに負荷を感じさせていた。

POLYPHONはそういった事態を解消させるべく開発されたボードである。そのため，ボードには，サブCPUが搭載され，このサブCPUで本来PCM8がする処理を，本体の68000とは並列に演算し，出力するという機能が付加された。

サブCPUはTMP68303。これはASSPといってMC68000-16MHz相当品をコアCPUに，各種の周辺デバイスをビルドインしたプロセッサである。そのため，MC68000に慣れたX68000ユーザーなら，指定のファンクションコールを利用すれば簡単にボード上のCPUにジョブを回すことができるという利点がある。

さらに世相を反映してか，スロット不足に悩む人のためにSIMMを使った拡張メモリ，数値演算プロセッサ，MIDI端子とありとあらゆる機能が詰め込まれた。

しかしながら，POLYPHONを製品紹介した当時は，肝心のプロセッサを使用するファンクションコールが未整理な状態であり，お世辞にも使いやすいといえない状態であった。ボード上に搭載されたD/A(PCM出力)は使用できる状態にあらず，機能は数%しか使用されていなかったといえるだろう。

ソフトウェア開発環境が整っていない状態では，さすがに拡張ボードの集合体以上の利用価値があるとはいえまい*2)。ユーザーはむしろ付加価値のほうをPOLYPHONに求めているのだから。

現在のPOLYPHON環境は，それらがある程度改善されているようである。今回はこれらを重点的に評価したい。

＊1）前の音が鳴っている最中でも次の音が鳴るという構造。本体標準のままでは，1音なので，次の音がキーオンされると当然，前の音がなくなってしまう。

＊2）あまりにパーソナルな次元を走っている人にはわからないかもしれないが，ハードとソフトは同時にできなければならないものである。潜在性能がいくらあっても，実際に使える面でチャチになってしまえば，お話にならない。ヘビーユーザーの団体にこれを落としたとしても資料が揃ったり，ファームウェア（ソフトとハードをつなぐ部分のものとでもいっておこう）が真っ当でなければ，お話にならないのだ。

サブプロセス

POLYPHONにMC68000相当品が搭載されているということは，POLYPHON自身にHuman68k，もしくはその相当品を搭載することで，OSのみに依存するようなソフトウェアならばオンボードで実行できるということがわかるだろう。

OSのみに依存するようなソフトウェアとは，一般にGCCなどのコンパイラ，アセンブラなどの類のもので，グラフィックなどを使用しないプログラムのことだ。グラフィックなどを使ったプログラムでも，パッチ当てで対応は可能だ。すでにJPEGローダなどへ対応が進んでいるようだ。

現在のPOLYPHONは，そういったプログラムをサブプロセスとして，本体のOSとは別に並列動作できる。このようなサブプロセス実行は，POLYPHONシステムというデバイスドライバを組み込むことで可能となる。

現在のPOLYPHON-24は24MHzと高速動作するので，プログラムのコンパイル，アセンブルなどをPOLYPHON側で高速に行えるということになる。

具体的にいうとコマンドラインからは，

A>PL ○○.x

のようにして起動する。

プログラムは完全に並列動作するので，コンパイル中，X680x0はまったくフリーな状態となる。それでなにをするかは，ユーザー次第。10MHz機ならばPOLYPHON側のほうが高速なので，転送のオーバーヘッド*3)を含めても，おそらくメインCPUでセコセココンパイルするよりも速い。また，このためのRAMは2Mバイト標準装備されている(SIMMとは違う)ので，本体のRAMを利用していないという事実も見逃せない。

プログラムをロードするモジュールなどもあり，資料もそれなりに揃い始めてきているから，アセンブラユーザーならモジュールを自作することも可能だろう。

しかし難点もある。

POLYPHONシステムとPOLYPHON用PCM8（つまりPCM8SB）はどうやら，同時に使えないようなのだ。ハード構成上，まったく不可能なことではないので（たとえばPOLYPHONシステム上でPCM8SBをタイムシェアリングさせるとか)，今後どうなるかわからないが，現在のところPOLYPHONシステムはデバイスドライバであるため，同一システムでの使い分けがしにくい。

また，新しいシステムにはPOLYPHONを効果的に利用したシステムが入っているが，これらの同時使用についてはあまり考えられてないようである。

ボード上のD/Aから，内蔵DMAに至るまで，セマフォ管理やタスク管理をしっか

りとしないと，あまり使いやすいボードとはいえないのではないだろうか？　まぁ，SXでさえ，そのあたりは難儀しているようなので，たいへんなのはわかっているのだが……。

＊3）メインMPU,MC680x0とサブASSP,TMP68303F間のデータ転送にかかる時間。実はこの時間も馬鹿にならないので，この手のボードでは設計では常に難儀な部分とされる（実は私も現在それで悩んでいる）。
POLYPHONは，そのものの価格の大半を占めているだろうと予想されるSRAMが多量に搭載されているので，オーバーヘッドはかなり少ないほうであるといえよう（予想）。

オンボードD/Aについて

POLYPHONには，D/Aコンバータが搭載されている。これを利用した楽器としての新展開を期待している方も多いだろう。

まず，このD/Aは，私が今月PPIボードで本体につけたものと似ている。データ形式も同じ8ビット符号なしというもの。厳密にはPOLYPHONの回路図を見ていないからわからないのだが，どうやら，付属プログラムを見ていると，このプログラムを実行したときは，POLYPHONシステムやPCM8SBなどに，POLYPHON上のタスクを占有されていると思われる。つまり，再生時には再生だけに一生懸命であるということ。

TMP68303にはDMAが内蔵されていて，このDMAの応用次第では，POLYPHONシステムやPCM8SBと同時に動かすこともできるだろう。しかし，そうするためにはボードもあらかじめそのように設計されていなくてはならないし，マネージャの作成も難しい。このあたりの情報は詳しく公開されていないのでわからないが，それくらいのポテンシャルを持っていそうなボードではある。

間違いなく標準となりつつあるPCM8のファンクションコールを意識し，サブボードを利用した音程変化などをうまく詰め込むには，多少難儀な部分があるかもしれない。要はソフト次第ということか。

また，ボード上にあるD/Aの質は，あまりほめられたものではない。確かに，純正のMSM6258V（AD PCM）よりもよい音ではある[*4)]。POLYPHONが出た当初ならばこれでよかったかもしれないが，現在ではもう役不足の感がある。玩具のようなチープな音ではユーザーは納得しないだろう。

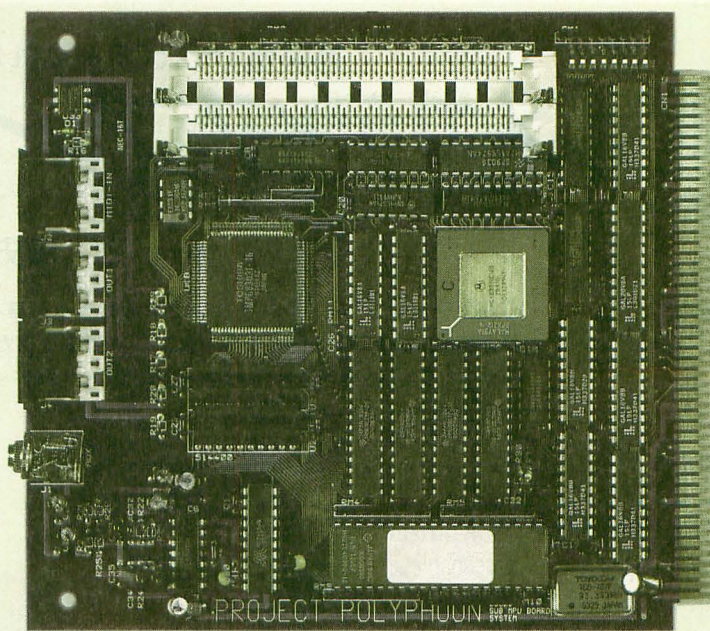
音の質は誌面では伝えられないが，マニアックな話にしてよいなら，システム2[*5)]程度の音。もっとも，システム2はサウンド用簡易DSPがなんらかの処理をしており，本来の音よりもよくなっているようなので，それよりもちょっと劣る。大きな違いではないが要はそんなところだ。

これでいいか否かは，個人差があるだろうが，先を見た設計というには，多少考えさせられる部分がある。扱うデータ量や処理速度，そしてコストとの兼ねあいもあるのだろうが……。

＊4）しかし分解能は本体AD PCMのほうが高かったりする。もっとも，周波数，非可逆である部分を考えると結果的にはPOLYPHON上のPCMのほうがよい音で再生されるのであろうが。
＊5）ナムコの90年代初期に活躍したゲーム基板。ワルキューレの伝説やオーダインなどがその基板で実現された。

POLYPHONはおいしいボードか？

私の目から見た場合，POLYPHONはSUPER以前のユーザーにはおいしいボードといえる。しかし，実質的には数値演算プロセッサの実装，メモリの増設などは，XVI以降のユーザーにはほとんど関係ない部分ともいえる。

いっそのこと完全なサブプロセッサとしての道を歩み，メモリ増設，演算プロセッサの増設など機能を除外して，もっと安いボードを作ったほうがよいのでは？　と思う。もっとも，私もボードを作る身だから，これらを除外しても，たいしたコストダウンにはならないとはわかっているのではあるが（ソケットの使いすぎもコストアップにつながっているのではないかな）。

POLYPHONは実に多彩な側面を持ったボードである。

スロット節約型拡張ボードとして見るとXVI以前，特に非SCSI機にはうれしいボードである。非標準といってもMIDIが16チャンネル増やせるならそれもいいだろう[*6)]。Z-MUSICの次のやつではあわせて80チャンネルできるだろうしね。

PCM8が軽くなるというだけで価値を見いだす人もいるかもしれない。Z-MUSICを使ったゲームなどでは，PCM8SBを組み込んでAD PCM多重再生をしている人もいるようだ。リニアPCM音源としては，システムとバランスは取れているが，クオリティには不満も残る。

そして，サブCPUボードとして見た場合，既存のコマンドをサブボード側で実行するだけでなく，並列実行のシステム化を進めていく必要があるように思われる。いずれにしてもこれはソフト次第である。

話はまったく変わるが，POLYPHONの添付ディスクは，あまり使いやすいとはいえないと思うぞ。最近のOh!Xのオマケディスクでも似たようなことがいえるから，大きなことはいえないけど，やっぱり，デフォルトで自動バッチを使って，サンプルのプログラムが動いたほうが楽でしょうに。結果的に中身のドキュメントをほとんどすべて読むまで利用方法がわからなかったぞ。

＊6 TMP68303内蔵のシリアルアウトを利用しているだけにすぎないから，純正コンパチにしてくれといわれてもできないのだ。RS-MIDIに近いものだから。仮にMIDIを取ったとしてもコネクタ代しか安くならないってわけ。

高解像でSXは夢を見れるか？

メガディスプレイへの道

Taki Yasushi 瀧 康史

CRTディスプレイ，それはパソコンのもっとも基本的な周辺機器です。高解像高性能なマルチスキャンディスプレイ駆使するためにCRTCの限界に挑みます。突き詰めればX68000究極の1024×1024ドット表示，1Mピクセルの世界が広がっているのです。

SX-WINDOWを使用していると，より美しい映像，広い画面がほしくなってきます。同時にメモリと処理速度も要求されてくるのですけど，後者の2つはX68030で，ある程度解決したといってよいでしょう。X68030なら処理速度は普通に使えば問題なく，メモリも12Mバイトあればおとなしく使っている分には問題にはなりません。

これにさらにSX-WINDOW ver.3.1で新たに増設されるような機能が加わると，さらにメモリがほしくなってくるのでしょうが，これはまあおいといて，前者をなんとかするような話を進めていきましょう。

それではまず，今回の内容を結論からお話しすることにします。

まずは，X680x0の性能を引き出すにはモニタを選ばなくてはいけないという事実です。そして，類のないX680x0の誇るべきCRTCの性能。そしてその効果的な引き出し方を修得すべきです。

本体無改造，または要改造で達成しうる高解像度の画面。最終目的は見やすい1024×768(ドット比1：1でのX680x0での最高画面)を目指して，自分のモニタの性能をできる限り引き出しながら，CRTCを操作する方法を暴いていくことにします。

モニタはまったく純正，プログラムはまったく組めない初心者から，アセンブラプログラマまで。特にアセンブラで自分でプログラムが書ける人には，理解すれば自分の思う最適の画面モードを作れるようになってもらおうと思います。

モニタの選択

広く美しい映像。それは，細かなドットの解像度，そして多くの色数を必要とします。よって，大画面で発色のよいモニタが必要になるでしょう。

正直にいうとX68000シリーズの純正モニタのデキはあまり誉められたものではありません。ですからここで明言しましょう。

X680x0の機能はモニタによって潰されている，と。

実際，X680x0にはもっといろいろな解像度モードがあります。しかしながら，これはモニタの性能によって制限されているのです。

実は私も先日新規にモニタを揃えまして，そのとき2,3ある事実に気がつきました。その事実を表にまとめてみたので，とりあえず表1をじっくりとご覧ください。

表のとおり，シャープはX680x0専用のモニタをいくつか出しています。表1での純正モニタは，代表的でかつ新規ユーザーがX680x0と一緒に買っていくものに注目した結果です。

旧ユーザーはわかるかもしれませんが，実はCZ-600Dから続くテレビつきの15k/24k/31kのモニタは表1に入っていません。実は，この上にCZ-615Dというものが存在していて，これこそCZ-600Dの後継といえるモニタなのですが，従来の3モードにVGAモード(640×480)を加えただけで，定価16万円を越えてしまう代物だったのです。実売価格が調べ切れなかったのが弱いのですが，おそらく安くても11万超ぐらいの価格で売られていると予想されます。

これでわかるとおり，純正モニタの選択は非常につらいものになってきています。

売れ筋……というかそれ以外の選択ができないといわれるCZ-607Dですが，これは24kHzを映すことができません。つまり，「68モニタを買えば，ゲーム基板やコンシューマゲーム機，PC-98シリーズまでたいていのものをつなぐことができる」という神話が崩れたともいえます。

次に値段を見てください。

「X680x0モニタはテレビが映らなくてはならない！」そういう固定概念を捨てろといっているわけではありません。しかしながら「テレビが映ること/15kHzでゲームが楽しめる」ことを捨てるだけで，モニタの選択権がずっと広がるのは事実です。

X680x0のゲームの大半は15kHz/31kHzの切り換えのものが多く，なかには24kHzで適当な大きさ（多少大きくなるケースが多い）にするソフトがあるものの，たいていは31kHzというモードを持っています(15kHzでしか映すことができないゲームがないわけではない)。

ここで仮に15kHzを捨てることにした場合どうなるでしょうか？

売れ筋のCZ-607Dと同じ値段で，三菱RD15D，IDEKのMF-8615などを選ぶことができます。IDEKは定価自体安いので，地方でもほぼ変わらない値段で買えることは強味でしょう。さらに国民機のWINDOWS化，DOS/Vマシンの普及にともなって，17インチディスプレイもすぐ手の届くところまできています。15インチでもよいならば，

表1　X680x0に接続可能なモニタの一覧

メーカー	型　番	水平/垂直同期	ピッチ/インチ	実売価格	備考
SHARP	CZ-607D	15.98k/61.4,31.5k/55.5	0.31/15	58,000	NTSCテレビモード有り
SHARP	CZ-608D	15.98k/61.4,31.5k/55.5	0.28/14	69,000	ノングレア
IDEK	MF-8615	24.8k〜69.0k/50〜120	0.28/15	58,800	ノングレア
IDEK	MF-8617	23.5k〜86.0k/50〜120	0.26/17	104,800	ノングレア
IDEK	MF-8517	23.5k〜86.0k/50〜120	0.28/17	94,800	ノングレア
SONY	CPD1438	24k〜58k/55〜115	0.25/14	69,800	トリニトロン
三菱	RD17Y形	24.8k,30k〜66k/50〜130	0.28/17	105,000	
三菱	RD15D	多分同上	謎/15	58,000	
NANAO	F557	24k〜65k/55〜90	0.28/17	105,000	フラットスクリーン
NANAO	F347II	24k?〜61.5k/55〜90	0.28/15	89,800	フラットスクリーン

注意）いずれも秋葉原価格です

かつては高嶺の花だったナナオのモニタなど，映りのよさは格別のSONYのトリニトロン管も射程範囲といえるのではないでしょうか。

カタログスペックではわからない，「色合いのよさ」「映りのよさ」などは，店頭でしか調べることができません。しかし，私が見た限りでは，これらのモニタと純正モニタでは格段の差があるように思えます。15kHzを諦めることによって，同じ値段でワンランク上のモニタを選ぶことができるのは見逃せません。かつてCZ-613Dなどを購入した値段とほぼ同じ値段で，いわゆる映りのよい17インチモニタが手にはいる時代になっているのです。

ひとつの選択に，15kHzはテレビに任せるという方法があります。

15kHzは主にゲームをワイドに楽しむ目的で使用されますが，電波新聞社から出ているXAV-1s(実売約6000円)を利用すれば15kHzRGBはビデオ端子，S端子に変換され，テレビに出力できます。

純正モニタはキーボードで簡単にコントロールでき，この利点はなかなかはずせないものがあるとは私も確かに思います。スーパーインポーズ機能，通称イメージユニット1(CZ-6VT1)との相性，それらは格別で，使いやすさからいえば純正がいちばんなのですが，少しの冒険と決心でまた違う選択ができることは，新規ユーザーには特に見逃せないことではないでしょうか？

画面同期信号の謎

取扱説明書のアナログRGB端子を見たことがある方はわかると思いますが，コンピュータが実際にディスプレイに画面を表示する場合，5つの信号が必要になります(基準値としてのGNDは除く)。

実際にはどのような信号があるのか見ていきましょう。まず，RGBの色をつけるための各R，G，Bの信号。そして水平のドットを置くタイミングを計る水平同期信号(H-SYNC)と垂直に画面が流れないようにするための垂直同期信号(V-SYNC)というものがあります。

シャープのモニタにはさらに，Audio R，Audio Lというものがあり，またYSという信号も存在します。

Audio L，Rは察しがよい方はすぐわかると思います。これはオーディオのAUX端子と同じものです。シャープ純正モニタのなかでは，いくつか耳のようなスピーカがついているものがあり，これらのための信号線といえるでしょう。

YSというのはコンピュータ信号の有無を示すもので，どうやらコンピュータの電源が入るとハイレベルになるだけの代物のようです。これを利用すれば，コンピュータと同時か，ほぼ同期して電源のON/OFFができそうな信号なのですが，なんにせよ，この信号はシャープのモニタ以外では見たことがないので無視することにします。

さて，モニタによっては，水平同期，垂直同期の2つの信号にXORをかけ，複合同期(SYNC)というものを作っているものもあります。こういった場合，実際の信号線はGNDを除いた4本になり，場合によってはこのSYNCをGREENに乗せ，シンクオングリーン(Sync on green)という3本の信号線でまとめてあるモニタもあります。

実際に画面を映すためには必然的にR，G，B，H-SYNC，V-SYNCの5つの信号が必要になるので，これらはモニタ内部でデコードされます。このうちRGB信号はアナログ信号ですがH-SYNC，V-SYNCはれっきとしたTTLレベルのデジタル信号です。

RGB信号がアナログですので，この信号にはドット情報は含まれていません。しかしながら，コンピュータではアナログ値になんとか区切りをつけ，1ドットの大きさを導き出さねばデータとして扱うことができません。

音声データの場合，音の振幅をサンプリングビット数で割り，音の進行をサンプリング周波数で割りました。同様にして，映像データの場合もデータのデジタル化（量子化）をしなくてはコンピュータで扱えませんから，ある程度のところで区切りをつけます。

すなわち音声データのビット数に当たる部分が色数のビット（X680x0では5ビット×RGB3色)になり，サンプリング周波数にあたる部分がドットクロックというわけです。このドットクロックは言葉どおり1ドットの表示にかかる時間を表しています。適当なドットクロックがここにあるとして，図1,2,3を見てください。

ドットの信号は普通，左上から右上に流れ，そのあと1段下に落ち，次のラインを左から右に流れ，最終的に縦のドット分流れ終わったあとに右下に到達し，また左上に戻ります（図1）。これを踏まえたうえで，まず図2を見てください。水平の場合と垂直の場合，同じように考えます。

まず水平です。

図中，水平データ表示期間とは実際1ライン水平に画面を表示する時間，つまり，

図1　ノンインタレース時の描画

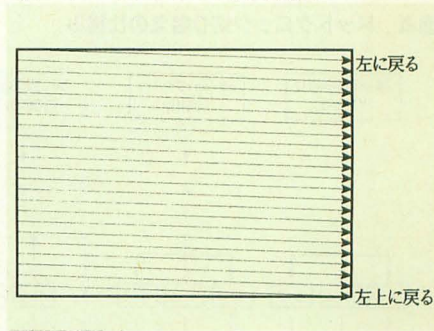



図2　CRTへの信号

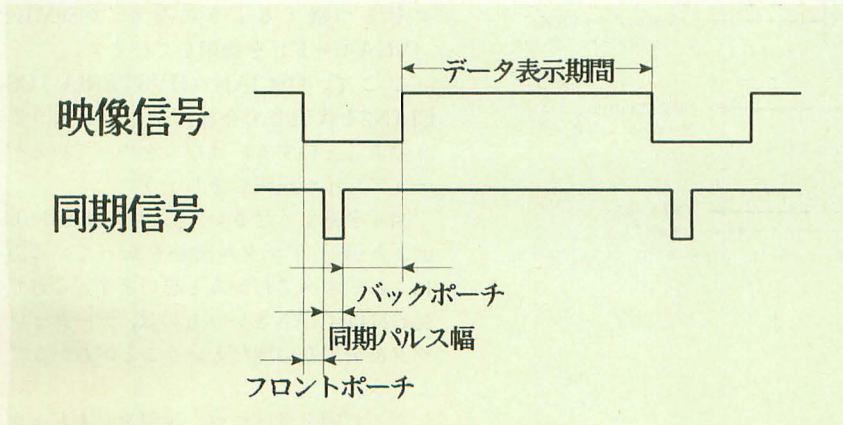



図3　実際のモニタとの相関図

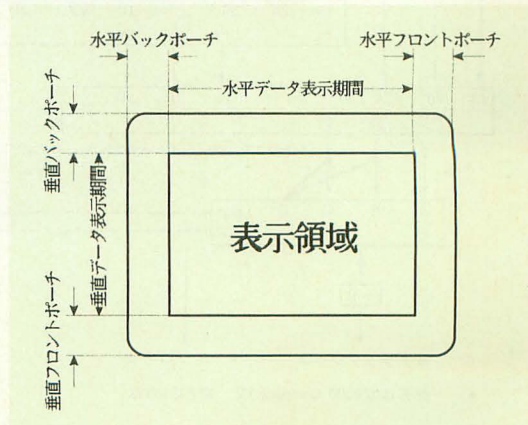



図3でいう表示領域の水平ドット幅でかかる時間のことです。当然ドットクロックから，

○水平データ表示期間＝
　横（水平）ドット数
　　×ドットクロック

という式が割り出せます。

図2より映像信号がLowになっているところは，表示していない時間です。表示していない＝ブランキングというわけで，次のような式が導き出せます。

○ブランキング時間＝
　フロントポーチ
　　＋同期パルス幅
　　＋バックポーチ

です。この式は，水平にも垂直にも適応します。さらにいうならば，

○同期期間＝
　ブランキング時間
　　＋データ表示期間

こういう式も導き出されることがすぐにわかるでしょう。これは図2からすべてわかることです。

さて，実際この信号はモニタ上でどう表記されているのでしょうか？　ここで画面の描画の実際を追ってみましょう。最初は水平データ表示期間です。これが表示し終わったあと，次にくるのは水平フロントポーチです。図3からわかるとおりこれは画面の右側のなにも写っていない部分です。水平フロントポーチが終わったあとは，図3では現れていませんが，図2には，水平同期パルス幅というものがあります。これが画面を左右に流れさせないようにするための同期信号です。

このあとは水平バックポーチ期間に入ります。図1上で見てわかるとおりモニタが1ライン表示すると，ひとつ下の段の左側から出てきますから，結果的には図3で表すように左側の黒い部分の幅になります。

このようにして水平同期期間は1周回るわけです。この時間を逆数にしたものが，一般にモニタの定格でいわれる「水平同期（たとえば31.5kHz)」というものなのです。

水平がすんだところで，垂直の説明をしましょう。水平同期では単位は常に1ドットクロックでした。垂直のこの単位にあたるものが，水平同期期間だというのはわかりますか？　要は縦は水平1ラインを基準に書くわけで，図1のとおりに書き進めるわけですから当然，単位となるのは水平同期期間なのです。

さて，この水平同期期間を踏まえたうえで，さっきの図2を見てみましょう。

○垂直データ表示期間＝
　縦ライン数
　　×水平同期期間

まずこの式が成り立ちます。これ以外はあとは水平の場合と同じで，あとは垂直バックポーチ，垂直フロントポーチがそれぞれ，画面の上，下になるだけです。垂直同期パルス幅は，画面を縦に安定させる信号ですので，これが短いということは画面が縦に流れてしまうという状態になるのです。

垂直同期期間は水平と同じく，ブランキング時間とデータ表示期間を足したものです。この逆数がモニタの定格で垂直同期といわれるものなのですね。

また，考えてみればわかりますが，垂直同期周波数というのは，1秒間に画面を書いている枚数ということになります。

テレビは約61Hzですから1秒間に約61枚画面を書き換えているというわけなのです。まぁインタレースですが（インタレースの説明は後ろの項で）。

図4　ドットクロック切り替えの仕組み

＊……部分はX68000 Compact XVI，X68030のみ。

OSCIAN/2，VICONとの関係

次はX680x0内蔵のCRTC，VICONとこのVICONにドットクロックを入力するためにパルス幅を整えるゲートアレイOSCIAN，OSCIAN2についての関連を説明しましょう。

前項でドットクロックが画面モードすべてに関連することは理解できたと思います。そのドットクロックはいったいどこでできて，結果的にどうやってCRTへ送るH-SYNC，V-SYNCを作り出しているのでしょう？

X680x0を一度ぐらい開けたことがあるかたならわかるとおり，ドットクロックは2，ないしは3個のオシレータから生成されています。原クロックを作るオシレータからは初期型はHYSTERIA，ACEからXVIまではOSCIAN，そしてCompactXVI以降，X68030に至るまでのOSCIAN2といった，ゲートアレイを経て，VICON（初期型はVINUS1,2）に流れ込みます。登載されているオシレータは，約39MHz，約69MHz，そしてOSCIAN2になって原クロックを3つ扱えるようになり，約50MHz（VGAモード）を使用しています。

ここで，OSCIAN（HYSTERIA，OSCIAN2を代表して今後区別がない限りこう呼ぶことにする）はなにをやっているゲートアレイか説明しましょう。

図4を見てください。（X68030Inside/Outより引用）デジタル回路を知っている方はこれですべてわかると思います。これで見ればOSCIANというものは，データセレクタ＆分周器の塊だということがわかるでしょう。

ここで例を挙げつつ，説明をしましょう。

まず我々は「SX-WINDOWでできる限り広い画面を作りたい！」そういう願望があります。ここでいきなり制限のありすぎるモニタでは少々難しいので，ここには仮に制限のまったくないモニタ(水平/垂直なんでも自動追従）があることにします。

表2を見ながら読んでください。

とりあえず広い画面がほしい。知っている人は知っているとおり，X680x0シリーズのテキスト画面の実画面は1024×1024ドットです。モニタの縦横比は（いろいろといわれてはいますが)，3：4といわれています。つまり，640×480はドット比（アスペクト比)が1：1ですし，1024×768も1：1，果ては1600×1200なんてのも1：1になるとわかると思います。ちなみに，768×512は2：3であり，モニタにあわせて（3：4）表示すると，グラフィックは多少縦に伸びてしまいます。アスペクト比を1：1にするには，縦を576ドットぐらいにしなくてはなりません。

話がそれてしまいましたが，アスペクト比が1：1になる最大のサイズというのは，横ドットの制限より，1024×768のサイズというのがわかると思います。

とりあえず，計算をしましょう。

当然ながら，ドットクロックが短いほうが，1ドットが小さくなりますから，それにあわせて，最短のドットクロックをみつけることにします。

再び図4をご覧ください。

ドットクロックオシレータの選択がまず入りますが，これは当然69MHzです。次に分周比を選びます。少ないほうがドットクロックが短くなるので(69MHzは周波数であり実際の長さの逆数であることをお忘れなく)，当然，÷2を選択します。最後の，÷8は無条件に入りますので注意が必要です。

すると，8ドットクロックは，

$$69\times10^6/2/8=4.3125\times10^6\ [\mathrm{Hz}]$$

となります。これは周波数ですので周期に直さねばなりません。ゆえに逆数をとって，

$$\mathrm{Fd}=1/4.312=0.23\mu\mathrm{s}/8$$

となるのです。つまりこの0.23μsという時間が，無改造のX680x0シリーズが生成できる最短の8ドットクロックということです。

8ドットクロックが画面のドットを水平に8つ書く時間ということは，すでにおわかりでしょう。VICON(CRTC)は，この8ドットクロックを基本にして，画面の同期に必要な信号を作り出しているのです。

この0.23μsのクロックを生むように，OSCIANのレジスタを設定します。

図4より，まず，HRLはAをセットしますから（÷2においてはA，Bどちらでも可ですが)，システムポートのHRLビットは0です。そして，HFビットはBを選ぶので，1です。最後にHDビットはCを選ぶので10となります。VDは垂直同期のところで出てくるのでいまはおあずけです。

これらの設定レジスタは図5，6に掲載されています（X68030Inside/Outより)。

水平データ表示期間は1ドットクロック×水平ドット数ですから，

水平データ表示期間＝

$$0.23\mu\mathrm{s}/8\times1024 = 29.44\mu\mathrm{s}$$

になることはもうおわかりでしょう。

とりあえず，表2の水平データ表示期間の欄を確認してください。

次に，同期パルス幅，バックポーチ，フロントポーチですが，これらはすべて経験とカンによって設定します。それぞれの設定について説明するので，前項の信号の意味を確認しながら読んでください。

まず，同期パルス幅ですが，これは必ず必要なものです。これが短すぎると同期が取れなくなり，左右に流れてしまい，また，長すぎると同期は取れますが，VRAMとのタイミング取りでプログラム側が失敗する恐れが起きてきます。

また，バックポーチは右側の空白部分です。これが短すぎると，同期信号と映像信号が混ざってしまい（というか，おそらく内部の信号遅延時間が間にあわなくなり，映像信号の立ち上がりを検出できなくなっていると思われる。推測)，色あいがグチャグチャになります。フロントポーチも同様です。

私の経験上，図2を見るだけで直感的にわかるようにしたつもりなのですが，フロントポーチと同期パルス幅がほぼ同じ時間で，バックポーチはその2倍から3倍の長さを取ったほうがよいと思われます（最大限ケチった場合)。

要は，ちょうどよく設定するのがコツなのですが，なかなかこれは経験が左右する

図5　CRTC R20のビット配置

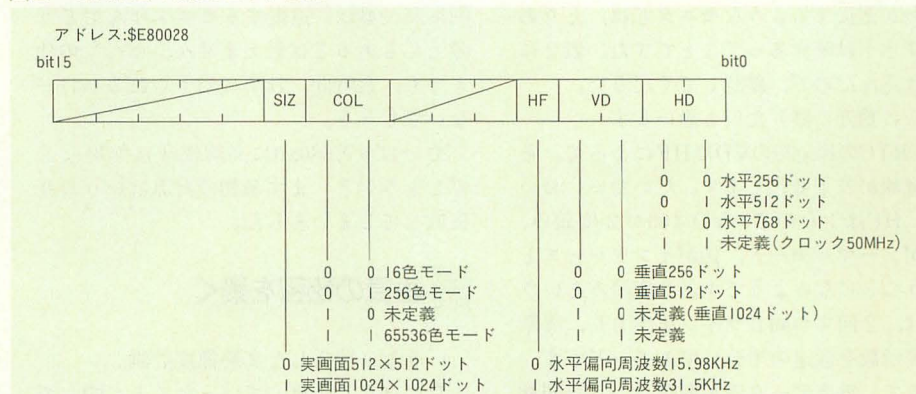



図6　システムポート#4のビット配置

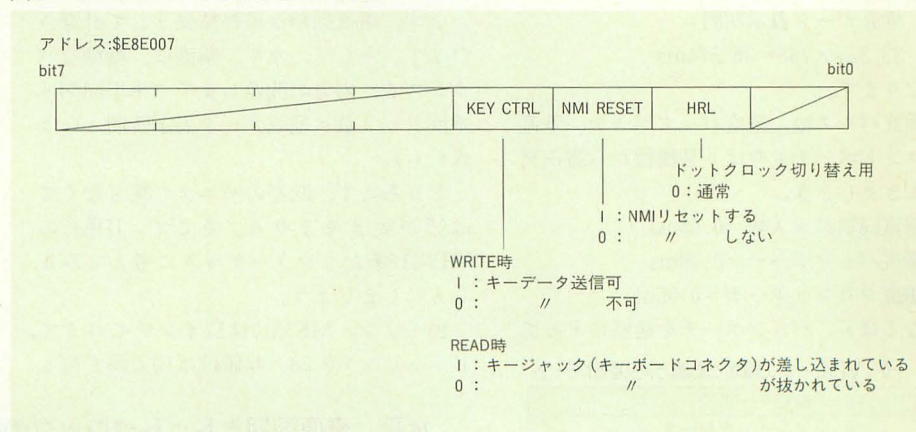



表2　1024×768モードをめざして

ドットクロックFd＝0.23μs　Reg20＝$16　HRL＝0

	同期期間	データ表示期間	同期パルス幅	バックポーチ	フロントポーチ
水平	33.3 μs/ 30.03 kHz	29.44 μs	1.2 μs	2.125 μs	0.5 μs
垂直	26.15ms/ 38.24 kHz	25.57 ms	0.05 ms	0.482 ms	0.05 ms

ものなので，一筋縄ではいかないのが事実です。こういうときは人の設定をとりあえず真似するのが無難でしょう。

他機種で1024×768モードをサポートしている某真金取主のこの3種類の信号を盗み見てきましょう。

チラリ……

どうやら，水平フロントポーチが0.5μs，水平同期パルス幅が1.2μs，水平バックポーチが，2.125μsのようです。水平同期パルス幅を大き目に取って，画面に余裕を与えているようですね。

では，これを真似して表2に書いてみることにします。それぞれ確認してください。

これらの合計時間が水平同期期間ですから，まとめて計算してしまいましょう。

水平同期期間＝
(29.44＋1.2＋2.125＋0.5)μs
＝33.3μs

約33.3μsになってしまいました。逆数を取ってみましょう。

どうやら30.30kHzです。つまり30.30kHzが追従するようなモニタでは，とりあえず水平は映せるってことですね。表2に書き込んだので，確認してください。

次に垂直に移りたいと思います。

CRTCのReg20のVDはHFによって，その意味が若干変化します。だいたいにおいて，HFは1なので，VDは00が2度読み，01がノーマルモード，10がインタレースということになるようです。2度読みというのは，2回ずつ同じラインを出力し，実際の表示数を設定の半分にするモードです。

さて，垂直データ表示期間は，水平同期期間のやはりドット倍ですから，

垂直データ表示期間＝
33.3μs×768＝25.574ms

になります。

垂直パルス幅，垂直バックポーチ，垂直フロントポーチもやはり某機種から盗み見してきましょう。

垂直同期パルス幅＝0.05ms
垂直バックポーチ＝0.48ms
垂直フロントポーチ＝0.05ms

なるほど。バックポーチを猛烈にとっていますね。それによって同期信号から映像信号までを開け，モニタを安定させるといったところですか。

これも利用して垂直同期期間（合計）を計算してみましょう。

垂直同期期間＝
(25.57＋0.05＋0.05＋0.48)ms
＝26.15ms

と結果が出ました。表2を確認してください。

これの逆数が垂直同期になります。計算してみると38.24Hzになっているようです……，ん？　38.24Hz？

1秒間に38回しか画面を書き換えていません。これでは仮にこういうモードを追従するモニタがあったにしても，猛烈にちらついてしまいます。

モニタの定格一覧を見てみましょう（表1）。

IDEKのモニタでさえ，垂直同期は50Hz以降。水平同期30kHzは割と妥当なようで，追従するモニタは山ほどありますが，垂直同期38.24Hzに追従するモニタなんてどう考えてもあるとは思えません。こんなの使えても，長時間，仕事に耐えられるわけがないでしょう。

やっぱりX680x0に高解像度は無理なのでしょうか？　まず最初の計算はいきなり失敗してしまいました。

垂直の秘密を暴く

いきなり挫折した高解像度計画。

なにげなく計算していったことに関して，ここで少し反省を踏まえて考えましょう。

まず，垂直同期は最終結果として計算されます。そして，水平，垂直は，絶妙なバランスで，両方が関連します（水平同期×垂直ドット数＝垂直データ表示時間という式から）。

とりあえず，既存のモニタで映らなくては話が始まりません。そこで，IDEKのMF5517あたりをターゲットに考えてみることにしましょう。

表1より，MF5517は17インチモニタで，ドットピッチ0.28。お値段は10万弱で割と手ごろ。性能も水平24.8kHz～57.0kHz，垂直50～90Hzとこのクラスでは平均的（コストから見れば上かな？）。

CRT同期に関する式一覧（コラム）の4式が指示するとおり，垂直は水平に影響され，水平はドットクロックに影響されます。すなわちすべてはドットクロックに影響されることがわかるでしょう。ここでドットクロックをなんとか狭める方法を考えつけば，すべては解決しますが，OSCIANの性能から判断すればわかるとおり，改造マシンにしない限り，これは不可能な話となってしまいます。

改造話題はこの先の項に任せておいて，とりあえず，無改造でできる範囲で考えてみることにしましょう。

まず，垂直同期は，なんとしてでも，50Hzまでに抑えたい。しかし，一般にハイレゾオートスキャンモニタといわれるなかで，ナナオなどは55Hzからしか映らないことになってますし（映るけど），そもそも50Hzまで落ちると随分，ちらつきが目立ってくるのでできれば避けたいところです。せっかくちらつかないようにと，よいモニタを買ったのですから，あまりちらつかせたくないものでしょう？

ここで目安として，できる限り垂直同期は55Hz～50Hzに抑えることにしましょう。できる限り55Hzという線でいくことにしておきます。こういう条件の下で話を進めましょう。

これらの逆数をとると，垂直同期期間がわかるはずです。要するに逆算をしたいということです。計算によると，

18.18ms～20.00ms

の間に垂直同期期間が収まればよいことがわかります。できる限り18.18msに近づけることを意識しなくては，ちらついてしまいます。

ここで，この値から逆算し，垂直バックポーチ，垂直フロントポーチ，垂直同期パルス幅を引き算してみることにします。前項で「ケチる場合」といったのはこういうことなのです。

ケチればその分，垂直データ表示期間が長く取れます。垂直にたくさんのドットを

図7　HF，VDビットの設定と垂直方向の動作モード

HF	VD	動作モード
0	00	ノンインタレース
	01	インタレース
	10	非公開（インタレース）
	11	非公開（インタレース）
1	00	二度読み
	01	ノンインタレース
	10	非公開（インタレース）
	11	非公開（インタレース）

水平，垂直同期とドットクロックの関係式

水平データ表示期間＝ドットクロック×水平ドット数
水平同期期間＝水平データ表示期間＋水平フロントポーチ＋水平バックポーチ＋水平同期パルス幅
垂直データ表示期間＝水平同期期間×垂直ドット数
垂直同期期間＝垂直データ表示期間＋垂直フロントポーチ＋垂直バックポーチ＋垂直同期パルス幅
上記4式で計算は足りるが，さらに追記するとすれば，
ブランキング（非表示）時間＝フロントポーチ＋バックポーチ＋同期パルス幅

並べる割に，ちらつきを抑えられます。

ブランキング時間の3つの項目は，モニタによって臨界点が（おそらく著しく）違います。あまりに短いと，限界の低いモニタでは追従しません。そこで一概にはいえないのですが，ある程度こちら側で予想して，ケチってみることにします。

先ほど使った垂直ブランキング期間の3項目，フロントポーチ，同期パルス幅，バックポーチの3つは，それぞれ，0.05ms，0.482ms，0.05msでした。大昔に中残光モニタとして紹介したCR-5500/7000では，同期パルス幅ならば0.30msまでケチることができました。おそらくたいていのモニタならばこの程度までケチることが可能だと思われます。V-HOLDの調整つまみがあるモニタで，調整しても流れてしまう場合は，多少垂直同期パルス幅を増やしてみてください（もちろん，がむしゃらに増やしても駄目。自分のモニタの定格はマニュアルを見て調べよう）。

そこで，先ほどの理想の垂直同期期間から垂直ブランキング期間を引いた値，すなわち，理想の垂直データ表示期間を求めてみることにします。

$$(18.18-0.05-0.05-0.3)\text{ms}=17.18\text{ms}$$

仮に水平をそのまま使うことを試みてみましょう。垂直データ表示期間を水平同期期間で割ればよいのですから，

$$(17.18\times10^{-3})/(33.3\times10^{-6})=515.9$$

計算結果より，垂直を55Hzにする場合，516ライン程度しか表示しきれません。

確かに通常モードから比べて，横に多少画面が広くなったものの，アスペクト比はかなり狂ってしまうため，さすがに使い難さは拭い去ることはできません。

仮に50Hzに妥協した場合どうなるでしょうか？　同様の計算をしてみると，

$$(20.00-0.05-0.05-0.3)\text{ms}=19.6\text{ms}$$

$$(19.60\times10^{-3})/(33.3\times10^{-6})=588.6$$

約588ライン表示できるわけで，先ほどよりも，72ラインほど余計に表示することが可能になりました。それでもやはり，ドット比1：1にはほど遠く，このモードもいまいち使おうとは思えません。

ところで，ここまで読んできた読者は，以下のことにすでに気がついたでしょうか？

●ちらつき（垂直同期）を代償に垂直の表示ドットを増やすことができる

個人的には，垂直同期はさすがに60Hzはないと，せっかくのモニタも台なしになる気がします。画面のちらつきが垂直同期に依存しているのは，すでにおわかりでしょう？

さて，望みの垂直同期から，垂直ドット数までの逆算を行いました。都合よく，ノンインタレースでアスペクト1：1に近くするには，水平同期まで逆算する必要があります。

水平同期は垂直同期と違い，たいてい余裕があります。その理由は，X680x0の設計が31.5kHz前後を前提に作っているからであり，かつ，一般に使用されるであろうモニタが24kHz〜50kHzと31.5kHzから随分帯域を持っているからでしょう（もっとも純正モニタは本当に30kHz〜33kHzぐらいまでしか追従しませんが）。

水平同期は垂直同期の基準となる分，多少でもケチれば，そのまま垂直ドット数，垂直同期に響いてきます。とするならば，やはり設定の仕方で大いに違ってくるはずです。

しかし，表2の水平ブランキング期間の値はすでにギリギリの値まで達しています。これ以上水平同期期間を減らすためには，横1024ドットは諦めるしかありません。

それならば妥協案として，ドットクロックはそのままで，縦横比を3：4にし，アスペクト比を1：1にしたときの最大を垂直50Hzから逆算し，水平と垂直のドット数を割り出してみようかと思いました。

この式は，単なる連立方程式なので省略します。要するに，水平，垂直の式に縦ドットx，横ドットyを代入し，縦横比の3：4から，x：y＝4：3という値を代入してもっとも適するであろう値から，割り切れる値を持ってきただけの話ですから。

結果，880×660という解像度に落ち着いてしまいましたが，それでも面積比では，約1.5倍にはなっているはずです。

設定値は表3のようになりました。

水平同期34.32kHz，垂直同期50Hzと，垂直同期が少なめですが，これでも安定しているようです。純正モニタでも一部同期が取れる機種がありますが，(例：CU21)水平同期がオーバーしていることもあり，増えた分だけどうしても画面からはみ出てしまいます。

H-SIZE，V-SIZEがあるモニタならば映すことができるかもしれませんが，残念ながら，編集部に残っていて，現在使えるモニタのロットには，そういうものはありませんでした（垂直振幅という形で垂直幅だけは変えられるのだが)。確かめようがないのでしかたないですね。これらは，各自自分のモニタで実験してください。

ところで，ここで下の事実に気がついたでしょうか？

●ドットクロックが固定しているとき，水平を増やすと垂直は減らすしかない

つまり，ドットクロックオシレータが69MHzのままでは，いくらモニタがよいものでも，ノンインタレースではこれ以上，望めないことを意味しているのです（あと，ほんの少しは広げられますが……)。

インタレースの謎

SX-WINDOWを使っている方で，SXWINの隠しオプション-Gを知っている方なら，一度ぐらい，24kHz，1024×848インタレースモードを体験したことがあるでしょう。

常人では耐えられないちらつきを感じると同時に，この広さでSX-WINDOWが使えたらどんなによいだろうか？　と，思ったに違いありません

私はSX-WINDOWをver.1.1のころから使用しています。当然，これはかなり古くからのSX愛好家だと自信を持っていえます。一時期1024×848ドットモードに憧れたのも事実で，フリッカーフィクサーなるハードウェアを作ろうか真面目に悩んだ時期もありました(垂直同期を2倍のスピードで送るような装置。単純にはできないことがわかりましたが)。

しかし，超長残光21インチモニタというものがそんなに手に入らないという事実から，私はCU21HDを購入し，サングラスを意図的につけたりしてちらつきを極力抑え，目を鍛え(？)，背景をできる限りちらつかないように変えて，なんとか使ってきました。

やはり次第に視力が低下し（20時間近くもあんな画面を見て目がおかしくならないほうが異常か？)，耐えられなくなって，ごく最近では816×544モードで妥協していました。

まぁ，よた話はさておき，ノーマルの

表3　880×660/標準CRT不可/無改造

ドットクロックFd＝0.23μs Reg20＝＄16　　HRL＝0

	同期期間	データ表示期間	同期パルス幅	バックポーチ	フロントポーチ
水平	29.14 μs/ 34.32 kHz	25.30 μs	1.20 μs	2.14 μs	0.50 μs
垂直	19.81 ms/ 50.00 kHz	19.23 ms	0.17 ms	0.58 ms	0.05 ms

1024×848のモードがいかに魅力的かつ,破壊的であるかはわかってもらえたと思います。

あのモードのせいでインタレースという言葉のイメージが悪くなっているに違いありません。こんなときにインタレースの話を出すと,結局1024×768は無理だったのでインタレースでお茶を濁そうとしてるか?と思われてしまっても無理はないでしょう。

ところで,いったい,インタレースがどういう仕組みかわかっている人はどれくらいいるでしょうか? インタレース時の同期信号との兼ねあいはX68030Inside/Outを見ていただくとして,ここでは結果のみお話しすることにします。

インタレースという画面モードは,基本的には,奇数ラインと偶数ラインを垂直同期期間に同期して交互に表示しているといえます。結果,インタレースでは1画面を書くのに2倍時間がかかることになり,かつ,2周期目の画像も,ほぼ同じ位置に描くため,ちらつくことになるのです。

インタレースの画面で有名なものはテレビです。テレビの垂直同期は約60Hzですが,インタレースなので2周期でひとつの画面を完成させます。

ところが,インタレースの割に,テレビ画面がちらつかないのは,隣接した色が近い(特に奇数ドットならば下,偶数ドットならば上のドット)ため,輝度の違いがあまりない分,ちらつきが抑えられているのです。

コンピュータは文字などを表示しているわけですから,偶数ラインと奇数ラインでは輝度が1と0とまったく反転することが多々あります。そのため,テレビに比べると画面がちらつくというわけなのです。その証拠に,X680x0では,15kHzの512×512インタレースモードで,偶数ラインと奇数ラインが近いような画像,たとえば,取り込み画像などを表示すると,ちらつきがわからなくなります。

ところで,我々がいま表示したいと思うものは,SX-WINDOWの画面です。ウィンドウフレームなどはわざとちらつくようにさせているのではないかと思うほど,1ラインごとの輝度が入れ替わっています。こういったときには,超長残光モニタを使えばよいのですが,超長残光モニタ自身,現在は流行っていないせいもあって,モニタそのものを手に入れることは,まず不可能だと思われます。

また,長残光モニタは一般に発色が悪く,赤,緑の色は比較的濃い色で出るものの,青が水色っぽい青になってしまい,グラフィックを多用するX680x0にはまったく向かないモニタとなっています(それにゲームにはまるっきり向かない)。

そういった背景上,SX-WINDOWにおいて,1024×848ドットモードはタブー視されてきました。もっともまともに使えるモニタで比較的手に入りやすかったものが,CR-7000(中残光,それもいまでは手に入れるのは難しいと思える)というエプソンの17インチモニタで,これでも,日中はちらつきが気になり,10時間を超えるタイピングなどをやっていると,目が死にそうになりました(そもそもそんなにしているのが悪いという説がある)。さらにいえば,個人的にPC-98XL2用の超長残光モニタで実験してみた経験がありますが,これでもちらつきがはっきりわかりました(こちらは使えないほどではない)。

CR-7000というモニタは,PC-486GRという同社のハイレゾマシン用のモニタでして,このマシンは1120×750ドットをインタレースで動かしています(PC-98XLなどと同様ですが,H98とは違います)。しかし,このインタレースモードはまったくちらつかないのです。

どうしてでしょうか?

答えは,実に,垂直同期にあるのです。

X680x0の垂直同期は,ROMを解析し,レジスタに叩くデータから垂直同期,水平同期を解析すると,1024×848ドットモードでは水平24.8kHz,垂直53.5Hzと,垂直がきわめて少ない条件で構成されていました。ノンインタレースでもちらつきのわかる53.5Hzをインタレースしてしまうのですから実際に描く時間は,1秒間に26~27枚程度になります。これならちらつくのも当たり前です。

対して,PC-486GRなどのハイレゾモードでは,同じインタレースでも,垂直が約80Hz。この違いは大きく,このため,中残光程度でもちらつきが少なくなるのです。

実際,垂直同期が80Hzクラスになると,インタレースだといわれるまで気がつかない場合があります。さすがに短残光モニタでは多少のちらつきが見えますが(探さないとわからない程度。μemacsのクロスカーソルがちらつきません),CR-7000あたりの中残光モニタでは,ちらつきがまったくわかりません。

ここでわかるとおり,同じインタレースでも,垂直同期を大きめに取ることにより,随分と変わることになります。前項で,ちらつきを代償に,縦ドット数が増えるといいましたが,インタレースでは如実に表れます。値の設定には十分気を遣う必要がありますが,これによる利益は大きいでしょう。

X680x0の場合,VDを変えることにより,インタレースモード,ノンインタレースモード,2度読み(31kHz縦256モードのような感じ)モードと切り換えることができます。インタレースの場合,垂直同期は当然半分しか確保しないで構いません。

すなわち,前項で垂直を55Hzにするには,1024×516ラインしか表示ができないといいました。ここでの垂直は55Hzですが,垂直ドット数をインタレース768ドット,つまり384にするなら,随分余裕が持てることになります。

このあたりでインタレースにおける事実をまとめておくことにしましょう。

●インタレースにおける事実

1.モニタは残光が残ったほうがよい。
2.垂直同期は多いほうがよい。
3.モニタの輝度は暗いほうがよい。
4.サングラスなどをすると随分変わる。
5.モニタのドットピッチは細かいほうがよい。
6.隣接した色が近似色だとあまりちらつかない。
7.水平同期にはまったく影響しない。

これらの事実を経て,インタレースモードで再度1024×768ドットに挑戦することにしましょう。

表4が計算結果です。

同じくドットクロックは0.23μs固定です。Reg20の値はインタレースにするため,$1aを設定します(VD=10)。HRLは0のままです。

水平は1024のままなので,表2とほとんど変わりません。むしろ,水平パルス幅を短くし,水平同期期間をケチっています。垂直データ表示期間は先ほどとは違い,384倍でよくなるので,余裕はずっと増えます。それにしたがい,フロントポーチ,バック

表4 1024×768インタレース/標準ディスプレイ不可/無改造

ドットクロックFd= 0.23μs Reg20= $1A HRL=0

	同期期間	データ表示期間	同期パルス幅	バックポーチ	フロントポーチ
水平	32.78 μs/ 30.51 kHz	29.44 μs	0.60 μs	2.14 μs	0.60 μs
垂直	12.93 ms/ 77.00 kHz	12.53 ms	0.05 ms	0.30 ms	0.05 ms

ポーチ，同期パルス幅を多少ケチに設定し，できる限り垂直同期を上げようと試みています。

結果，水平同期30.51kHz，垂直同期77Hzと，マルチスキャンモニタならばたいていは同期が取れるような値になりました。

インタレースでも77Hzなので，中残光ならばちらつきが，ほとんどわからない程度，短残光モニタでもあまり気にならない程度になっているはずです。

これでとりあえず，ノンインタレースで880×660ドット，インタレースで1024×768ドットのモードができあがりました。

レジスタへのセット

CRTモードを変えるときに密接に関わりがあるCRTCレジスタは，Reg00～Reg08，Reg20，そしてシステムポート#4のHRLビットです。

このうち，Reg20はすでに設定方法をお話ししましたから（OSCIANの設定なので），ここではReg00～Reg08までのセット方法を説明しましょう。

まず，表5を見てください。

表5はいままで求めた結果をもとにレジスタの値を算出する式の一覧です。たとえば，表5の式に，表3の880×660モードの値を代入するとレジスタの値は，

・Reg00＝$007e
・Reg01＝$0004
・Reg02＝$000a
・Reg03＝$0078
・Reg04＝$02ad
・Reg05＝$0007
・Reg06＝$0018
・Reg07＝$02ac

と算出できます。Reg08は水平アジャストといって，デフォルトのままで設定しなくても構わないでしょう。

レジスタのアドレスは，表6に掲載しておきます。

実際のレジスタのセットについては，リスト2，SX-WINDOWのスタートアップメンテの登録プログラムを参考にしてもらうとわかりやすいと思います。アセンブラで記述するときに注意することは，Reg0～Reg08まではそのままmove.w転送し，Reg20は実際には$e80029にmove.bと，バイトで転送してCRTCの設定をするのが無難だと思われます。というのも，Reg20の上位バイトには，実画面モードのセット，色モードのセットなどのビットがあるからです。

いじって遊んでみるのも面白いでしょうが，SX-WINDOW上から実行するときには，下位バイトだけに書き込んだほうが無難だといえるでしょう。

また，システムポート#4には，CRTCとはまったく関係がない機能まで配置されています。HRL信号以外を動かすと，キーボードがまったくきかなくなったり，インタラプトスイッチ(NMI)がかからなくなったりするので注意してください（もっとも高解像度で設定しているときはHRLはほとんど0固定でしょうが……）。

Reg20の値設定は，データセレクタとの兼ねあいで，ややこしい限りです。実のところ，値はある程度決まっているので，表にしてみました（表7）。

HRLビットの兼ねあい，ドットクロックとの関係など，うまくまとめてあると思うので役立ててみてください。

また，表8に計算するための基本フォーマットを用意しました。これも関数電卓で同期期間を計算するときに役立つと思われます。

モニタによる臨界点の違い

最近のモニタはオートスキャンモニタなので，あまり考えなくて済みますが，X680x0ユーザーが一般に使っていると思われるCZシリーズ，CUシリーズには，やっぱり臨界点が存在します。表7で私はわざと，15kHzモードとか，31kHzモードという表記は使用しませんでした。理由は簡単で，それらはやはりドットクロックから計算されるものだからです。

IOCSのCRTMODに登録されている標準でサポートしているモードも実はすべて31kHz，15kHzとまとまっているわけでは

表6 CRTCレジスタへのセットアドレス

レジスタ	アドレス
Reg00	$e80000.w
Reg01	$e80002.w
Reg02	$e80004.w
Reg03	$e80006.w
Reg04	$e80008.w
Reg05	$e8000a.w
Reg06	$e8000c.w
Reg07	$e8000e.w
Reg08	$e80010.w
Reg20	$e80028.w
SystemPort4	$e8e007

表5 水平/垂直同期からのレジスタ計算式

$$Reg0=\frac{水平同期期間}{8ドットクロック}-1=$$

$$Reg1=\frac{水平同期パルス幅}{8ドットクロック}-1=$$

$$Reg2=\frac{水平同期パルス幅+水平バックポーチ}{8ドットクロック}-5=$$

$$Reg3=\frac{水平同期期間-水平フロントポーチ}{8ドットクロック}-5=$$

$$Reg4=\frac{垂直同期期間}{水平同期期間}-1=$$

$$Reg5=\frac{垂直同期パルス幅}{水平同期期間}-1=$$

$$Reg6=\frac{垂直同期パルス幅+垂直バックポーチ}{水平同期期間}-1=$$

$$Reg7=\frac{垂直同期期間-垂直フロントポーチ}{水平同期期間}-1=$$

表7 R20と8ドットクロックの相関関係

Reg20	HRL	8dc(μs)	垂直	分周計算
$00	0	1.647	ノンインタレース	39M/64
$01	0	0.823	ノンインタレース	39M/32
$04	0	1.647	インタレース	39M/64
$05	0	0.823	インタレース	39M/32
$10	0	0.690	二度読み	69M/48
$10	1	0.920	二度読み	69M/64
$11	0	0.345	二度読み	69M/24
$11	1	0.460	二度読み	69M/32
$12	0	0.230	二度読み	69M/16
$13	0	0.317	二度読み	50M/16(C以降)
$14	0	0.690	ノンインタレース	69M/48
$14	1	0.920	ノンインタレース	69M/64
$15	0	0.345	ノンインタレース	69M/24
$15	1	0.460	ノンインタレース	69M/32
$16	0	0.230	ノンインタレース	69M/16
$17	0	0.317	ノンインタレース	50M/16(C以降)
$18	0	0.690	インタレース	69M/48
$18	1	0.920	インタレース	69M/64
$19	0	0.345	インタレース	69M/24
$19	1	0.460	インタレース	69M/32
$1A	0	0.230	インタレース	69M/16
$1B	0	0.317	インタレース	50M/16(C以降)

ありません。それぞれにあわせて，少しずつ変わっています。

強引にあわせるときには，ブランキング時間である程度補正はできますから問題はありませんが，モニタの持っている100%ないしは120%の力を引き出すには，自分のモニタがどこまで追従するか確かめてみると面白いと思います。しかしあまり臨界点で使いすぎると，壊れるという噂もあるので各自の責任でやりましょう。

純正モニタでも，CU21は割と臨界点が高かったりしますし，CZ-6??シリーズでも，ものによって，もしくはロットによって違うかもしれません。CU21は水平同期は31.5kHz，24.8kHz，15.9kHzの前後2～3kHzを追従し，垂直は50～63Hzぐらいまで追従してくれますが，逆に同期パルス幅が短いとすぐに流れてしまうという欠点があります。

さて，これらは読者への課題ということにしておきますが，前項では，横1024ドットが追従すれば表示されるものとして仮定していましたが，実はそうでもありません。

それはH-SIZEです。

さすがに8ドットクロックがたったの0.23μsしかありませんから，どうしても横ドット表示に時間がかかってしまいます。

たとえば，真金取主などの1024×768モードでは，8ドットクロックは0.10μsですし，適当な国民機のウィンドウアクセラレータボードで，1024×768表示できるボードドットクロックを調べてみても，0.107μsなのです。1024表示するようなモニタはたいてい，0.10μs程度であると思われます。

そこを，我々は0.23μsで行うのですから，どこかに支障が出てもおかしくはありません。

横1024ドット表示するようなマシンは，通常，水平同期も50kHz～60kHzほどで画面を出力しています。我々は33kHz程度です。絶対的な水平データ表示期間が長すぎるのです。

つまり，モニタによっては，インタレース1024×768モードで左右を縮小しきれず，大きく画面からオーバーしてしまいます。CU21も，880×660モードが追従こそすれ，水平サイズが広すぎ（このモニタにはH-SIZEが全面パネルに存在しないため）結局表示し切れなくなって，使えないモードになっています。

サンプルは誌面の許すかぎりできるだけ多く載せますが，どのモニタでちゃんと1024×768のモードが映るかはこちらでは保証しかねます。

本体改造によるさらなる高解像

ここまで読んできてくれた読者ならば，CRTCのベースになるドットクロックが，いかに画面モードを左右しているかわかるでしょう。

最短8ドットクロックは，ドットクロックオシレータ/16なのですから，ドットクロックオシレータ69MHzを速いものに代えることにより，水平，垂直に余裕ができ，美しく画面が映ることになります。

たとえば，69MHzのオシレータを80MHzに上げるとしましょう。

8ドットクロックは0.20μsに下がります。たったの0.03μsだと思わないでください。仮に，横768ドット表示するならば，2.9μsも水平同期を稼ぐことができます。

結果的には垂直512ドット表示した場合，垂直も1.485msを稼ぐことができます。31.5kHz，60Hzの768×512モードは，約34kHz，約66Hzへと変わり，ちらつきがかなりなくなってきます。

また，横に1024ドット表示した場合には，水平だけで3.84μsの短縮となり，水平表示幅が画面からはみ出てしまうような場合でも，（その分小さくなるので）画面内に収まる可能性が出てきます。垂直にしても，1.475msの短縮は，垂直同期の向上につながり，ちらつきをかなり抑えることができるようになるでしょう。

100MHzのオシレータを入れた場合どうなるでしょうか？　8ドットクロックは実に0.16μsまで下がります。もうこれ以上は説明しなくてもわかりますよね？

改造は，ドットクロックオシレータである69MHz（拡張スロットの下についているもの）を100MHzの同サイズのものに変えるだけです。要はクロックアップのときと似たような感じです。ただ，かなり速いクロックですので，データセレクタなどを使って切り換えなどをしようとすると，それだけ信号が乱れてきます。当然ですけど，マシンクロックとは違い，目で見てわかるので，それなりの注意は必要でしょうね。

オシレータを抜くときも，マシンクロックオシレータとは違い，ランドが太いのではずすのにもテクニックがいります。幸い，マイナスドライバーが入りやすい場所にあるので，少しずつ確実に温めてはずしてください。スルーホールなどがはずれてしまうと結構面倒なので，取りはずしにも十分気を遣ってくださいね。

さらに注意すべき点は，オシレータの大きさです。CompactXVI以降はハーフサイズ（正方形）それ以前はフルサイズ（？）ですので，まず自分で本体を開け，大きさを確かめてから購入しましょう。

表8　モニタの同期計算のスケルトン

ドットクロックFd=　　μs Reg20= $　　　HRL=

	同期期間	データ表示期間	同期パルス幅	バックポーチ	フロントポーチ
水平	μs/　　kHz	μs	μs	μs	μs
垂直	ms/　　kHz	ms	ms	ms	ms

$$Reg0=\frac{水平同期期間}{8ドットクロック}-1=$$

$$Reg1=\frac{水平同期パルス幅}{8ドットクロック}-1=$$

$$Reg2=\frac{水平同期パルス幅+水平バックポーチ}{8ドットクロック}-5=$$

$$Reg3=\frac{水平同期期間-水平フロントポーチ}{8ドットクロック}-5=$$

$$Reg4=\frac{垂直同期期間}{水平同期期間}-1=$$

$$Reg5=\frac{垂直同期パルス幅}{水平同期期間}-1=$$

$$Reg6=\frac{垂直同期パルス幅+垂直バックポーチ}{水平同期期間}-1=$$

$$Reg7=\frac{垂直同期期間-垂直フロントポーチ}{水平同期期間}-1=$$

また，私はOSCIAN，OSCIAN2，VICONがクロックアップに追従していることは確認していますが，初期型のHYSTERIA，VINUS1，VINUS2が追従するかは，まだ未調査……というか，もう調査できる環境とはいいがたいところにいます。

それらの改造の責任は自分で取って行ってください（ほかの改造も結局は自己責任ですが)。私は編集部に毎日いるわけではないので，編集部に電話していただいても，まず120％わからないでしょう。

データ集とサンプル

ドットクロックをいろいろ変えたときのデータ集を表8に掲載しておきます。すべてのモニタで動くわけではないですし，動くモニタはなに？　といわれても困ります（これも編集部に電話していただいてもやっぱりわからないでしょう)。

最低限，水平同期，垂直同期の情報を掲載しておくので，自分で店員によく聞き，いきなり「マシンはなんですか？」と聞くような店員は無視してカタログを要求し，自分自身の力で映りそうなモニタを探してください。

仮に映らなかったとしても，そのモニタ専用の値を自分で作成してみるのも一興でしょう。

さて，サンプルプログラムです。

リスト1は，CRTのレジスタを直接書くことができるプログラムです。簡易的なものなので，面白いモードができたら，エイリアスやバッチファイルを駆使してみるとよいかもしれませんね。

また，リスト2は，SX-WINDOWのスタートアップメンテでの登録ツールです。編集部ではADJUST設定をうまく行い，1024×768モードでもちゃんとスクロール，最大化，FPウィンドウがちゃんとした位置に出るものができてはいるのですが，さすがにリストがひたすら長いので，掲載ができません。

とりあえず，単にCRTCのレジスタをいじるだけのものを作ったので，適当に自分で改変し，-Gオプションで近い値などをセットして，各自工夫をこらしてください。

まとめ

ふぅ。長い戦いでした。調べている間は楽しかったのですが，いかに説明すればいいか，すごく悩みました。はっきりいって。

今月は編集部の首都高速が見えるこの席で，何度，朝焼けにきらめく水面と，夕焼けにきらめく水面を見たことか。

しかしですね。

苦労したけれど，これで，ハイレゾ化とかという要望はある程度こなせましたよね。こうなってくると，OSCIANの最後の8分周を4分周にするモードとか，そういうのがほしくなっちゃうけど。あとはVRAMが2048×2048の実画面があれば完璧なんですけどね。しかし，無改造でこれだけの解像度が出るとは，確かに先を見たハードではあったようですね。

＊　　＊　　＊

さて，これはローテク実験室ではないですが，来月のローテクでは，引き続いてモニタ支援回路を作ろうかと思っています。とりあえず……いっちゃおうかと思ったけど内緒にしよう。楽しみは取っておいたほうがよいですからね。

できるならば，CRTC関連の質問は封書でお願いします。編集部内，瀧宛てで届くはずなので。答えられる質問ならば，誌面が許す限り，なんらかの場所で答えてあげてもよいですが，誌面が許してくれるかは私の関与するところではありません。ご了承ください。

参考文献）

Inside X68000
Outside X68000
X68030Inside/Out
NANAO T660iJのマニュアル
CR-7000のマニュアル

リスト1

```
  1: *$Author: Kohju $
  2: *$Header: a:/usr/home/Kohju/labo/crt/crtxf2.has,v 1.2 1994/04/16 17:24:08 K
ohju Exp $
  3: *$Log: crtxf2.has,v $
  4: *Revision 1.2  1994/04/16  17:24:08  Kohju
  5: *・R20レジスタの値設定を間違えていたのでFix.
  6: *
  7: *Revision 1.1  1994/04/16  15:53:46  Kohju
  8: *Initial revision
  9: *
 10:
 11: *
 12: *   CR-7000専用
 13: *
 14:
 15: *   注意）.rファイルに出来るけどしないでね。
 16:
 17: *   CRT定義プログラム
 18:
 19:     .include        doscall.mac
 20:     .include        iocscall.mac
 21:
 22: *   I/Oアドレス設定
 23:
 24: CRTC                equ     $00e80000                   * CRTCのアドレス
 25:
 26:     .text
 27:
 28: mdhead:
 29:     .dc.l   'OBJR'                              * モジュールヘッダ
 30:     .dc.l   0                                   * プログラムエリアサイズ
 31:                                                 * Xファイルなので意味無し。
 32:     .dc.l   $00000038                           * スタートアドレスオフセット
 33:     .dc.l   $00000412                           * ワークエリアサイズ
 34:     .dc.l   0,0,0,0                             * システム予約
 35: L000020:
 36:     move.l  a1,d2
 37:     lea.l   $0412(a1),a1
 38:     movea.l a1,a7
 39:     clr.l   -(a7)
 40:     .dc.w   $a350
 41:     move.l  $0008(a0),-(a7)
 42:     pea.l   (a1)
 43:     lea.l   $000c(a7),a7
 44:     movea.l d2,a1
 45:     movea.l a1,a5
 46:     move.l  a2,$0000(a5)
 47:     move.l  a3,$0004(a5)
 48:     clr.w   -(a7)
 49:     clr.l   -(a7)
 50:     pea.l   $0008(a5)
 51:     pea.l   (a2)
 52:     .dc.w   $a3ea
 53:     lea.l   $000e(a7),a7
 54:     move.w  d0,$0010(a5)
 55:     bsr.w   _INIT
 56:     bmi.w   L00005e
 57: L00005e:
 58:     bsr.w   _TINI
 59:     move.w  d0,-(a7)
 60:     .dc.w   $a352
 61:
 62: _INIT:
 63:
 64:
 65:     clr.l   -(sp)
 66:     DOS     _SUPER
 67:     addq.l  #$4,sp
 68:
 69:     lea.l   CRTC,a0                     * CRTC のアドレス
 70: *                                       * 30.51kHz/74.00Hz 1024*800 DotMode Inte
r
 71:     move.w  #$008e,$00(a0)              * Reg0
 72:     move.w  #$0002,$02(a0)              * Reg1
 73:     move.w  #$0007,$01(a0)              * Reg2
 74:     move.w  #$0087,$06(a0)              * Reg3
 75:     move.w  #$019b,$08(a0)              * Reg4
 76:     move.w  #$0001,$0a(a0)              * Reg5
 77:     move.w  #$000a,$0c(a0)              * Reg6
 78:     move.w  #$019a,$0e(a0)              * Reg7
 79:     move.w  #$001b,$10(a0)              * Reg8
 80:     move.b  #$1a,$29(a0)                * Reg20
 81:
 82:     move.l  d0,-(sp)
 83:     DOS     _SUPER
 84:     addq.l  #$4,sp
 85:
 86:     .dc.w   $a08f
 87:     move.l  d0,d2
 88:     move.l  d0,-(a7)
 89:     .dc.w   $a092
 90:     moveq.l #$01,d1
 91:     or.l    d1,d0
 92:     move.l  d0,-(a7)
 93:     move.l  d2,-(a7)
 94:     .dc.w   $a093
 95:     lea.l   $000c(a7),a7
 96:     rts
 97:
 98: _TINI:
 99:     rts
100:
101:
102: L000080:
103:
104:     .end    L000020
105:
```

リスト2

```
 1: /*
 2:  $Author: Kohju $
 3:  $Header: a:/usr/home/Kohju/Labo/crtregset/crtregset.c,v 1.1 1994/04/20 11:42:57 Kohju Exp
$
 4:  $Log: crtregset.c,v $
 5:  * Revision 1.1  1994/04/20  11:42:57  Kohju
 6:  * Initial revision
 7:  *
 8:
 9:  プログラムはgccおよび、libc用です。xcではコンパイルが出来ません。
10:  環境変数GCC_OPTIONに+が設定されている必要があります。
11:
12:  $ gcc crtregset.c -ldos
13:  で、コンパイル可能。
14: */
15:
16: #include <stdio.h>
17: #include <stdlib.h>
18: #include <sys/dos.h>
19:
20: void  main( int argc, char **argv )
21: {
22:   printf(  "crtregset.x $Id: crtregset.c,v 1.1 1994/04/20 11:42:57 Kohju Exp $¥n"
23:       "Free software,Presented by 瀧 康史¥n");
24:   if( argc!=12 ) {
25:     printf(  "¥n機能：CRTCレジスタに引き数の値をセットします¥n¥n"
26:         "書式) crtregset.x reg0 reg1 reg2 reg3 reg4 reg5 reg6 reg7 reg8 reg20 HRL¥n¥n"
27:         "説明) reg0～reg20 : CRTCのレジスタ設定です。16進数表記。¥n"
28:         "       HRL         : システムポート＃４のＨＲＬビットです。０か１。¥n¥n"
29:         "注意) 値はそのままCRTCに格納します。値チェックはしていないので、¥n"
30:         "       このプログラムよって、暴走させることも可能なので注意。¥n¥n"
31:         "       すべての引き数は省略出来ません。(手抜きのため) ¥n¥n"
32:         "エラーが発生しました。引き数が足りません¥n");
33:     exit(-1);}
34:
35:   _dos_super(0); // Supervisorモードに以降
36:
37:   *(unsigned short *)0xe80000=(unsigned short)strtol(argv[ 1],NULL,16); // reg00
38:   *(unsigned short *)0xe80002=(unsigned short)strtol(argv[ 2],NULL,16); // reg01
39:   *(unsigned short *)0xe80004=(unsigned short)strtol(argv[ 3],NULL,16); // reg02
40:   *(unsigned short *)0xe80006=(unsigned short)strtol(argv[ 4],NULL,16); // reg03
41:   *(unsigned short *)0xe80008=(unsigned short)strtol(argv[ 5],NULL,16); // reg04
42:   *(unsigned short *)0xe8000a=(unsigned short)strtol(argv[ 6],NULL,16); // reg05
43:   *(unsigned short *)0xe8000c=(unsigned short)strtol(argv[ 7],NULL,16); // reg06
44:   *(unsigned short *)0xe8000e=(unsigned short)strtol(argv[ 8],NULL,16); // reg07
45:   *(unsigned short *)0xe80010=(unsigned short)strtol(argv[ 9],NULL,16); // reg08
46:   *(unsigned char  *)0xe80029=(unsigned char )strtol(argv[10],NULL,16); // reg20
47:
48:   if((int)strtol(argv[11],NULL,16)==1){  // HRL
49:     asm volatile(          // 1
50:       "¥tmove.b $e8e007,d0¥n"
51:       "¥tandi.b #$fd,d0¥n"
52:       "¥tor.b   #$1,d0¥n"
53:       "¥tmove.b d0,$e8e007¥n");
54:   }
55:   else asm volatile(          // 0
56:       "¥tmove.b $e8e007,d0¥n"
57:       "¥tandi.b #$fd,d0¥n"
58:       "¥tmove.b d0,$e8e007¥n");
59: }
```

表9 データシート

説明：

実際に書き込むデータと設定している値の一覧表なので，各自モニタに合わせて適当に操作していただきたい。インタレースのちらつき度は，垂直同期を見ること。目安としては，70Hzを超えれば十分問題がない程度。ノンインタならば，50Hzに近くなるとかなりちらつく。50Hzを割ると長時間見ていられない。ノンインタのちらつきは，結構くるものがあるぞ。

また，ドットクロックオシレータが80MHz，100MHzと書いてあるものは，オシレータを取り換えたときのモード。速いオシレータに変えても，ハイレゾオートスキャンモニタならばちゃんとIOCS_CRTMODに登録されているものもそのまま映るので問題はない（ゲームとかしても大丈夫だよ）。

さて，これらの表はcrtregset.xで出力したログを加工したもの。

したがって，レジスタからの逆算値であり，データには計算誤差が含まれている。特に表示ドット数は，計算アルゴリズム上，実際よりも少なめになっているようだ。本文中に書き忘れていたが，すべてのCRTモードはフロントパネルの調整を，各自で行うこと。調整せずに映らないと嘆くと損するぞ。

1）表示ドット）横 879　縦 660　　ノンインタレース
ドットクロックオシレータ＝69.551MHz　　8ドットクロック＝0.23μsec

	同期期間	データ表示期間	同期パルス幅	バックポーチ	フロントポーチ
水平	28.99μs/34.5kHz	25.31 μs	1.15 μs	2.07 μs	0.46 μs
垂直	19.88ms/50.29Hz	19.13 ms	0.23 ms	0.46 ms	0.06 ms

レジスタ	0	1	2	3	4	5	6	7	8	20	HRL
データ	7D	4	9	77	2AD	7	17	2AB	1B	16	0

推奨モニタ）MF-8617,CR-7000他

備考）CU21でも一応追従する
本文中で利用したもの

2）表示ドット）横 879　縦 659　　ノンインタレース
ドットクロックオシレータ＝69.551MHz　　8ドットクロック＝0.23μsec

	同期期間	データ表示期間	同期パルス幅	バックポーチ	フロントポーチ
水平	28.30μs/35.3kHz	25.31 μs	0.46 μs	2.07 μs	0.46 μs
垂直	19.07ms/52.43Hz	18.68 ms	0.03 ms	0.31 ms	0.06 ms

レジスタ	0	1	2	3	4	5	6	7	8	20	HRL
データ	7A	1	6	74	2A1	0	B	29F	1B	16	0

推奨モニタ）F557,MF-8617,CR-7000他

備考）CU21を表示できない。その分，垂直同期を上げた

3）表示ドット）横 815　縦 620　　ノンインタレース
ドットクロックオシレータ＝69.551MHz　　8ドットクロック＝0.23μsec

	同期期間	データ表示期間	同期パルス幅	バックポーチ	フロントポーチ
水平	31.75μs/31.5kHz	23.46 μs	3.45 μs	3.22 μs	1.61 μs
垂直	20.35ms/49.14Hz	19.68 ms	0.19 ms	0.48 ms	0.00 ms

レジスタ	0	1	2	3	4	5	6	7	8	20	HRL
データ	8A	E	18	7F	280	5	14	280	1B	16	0

推奨モニタ）CZ-6xxD他

備考）純正モニタでのほぼ限界
CU21ではV-SIZEを最短にしないと上下がロールしてしまう
モニタによっては左右が多少切れる。ちらつくので覚悟すること

4）表示ドット）横 815　縦 544　　ノンインタレース
ドットクロックオシレータ＝69.551MHz　　8ドットクロック＝0.23μsec

	同期期間	データ表示期間	同期パルス幅	バックポーチ	フロントポーチ
水平	31.75μs/31.5kHz	23.46 μs	3.45 μs	4.37 μs	0.46 μs
垂直	18.03ms/55.46Hz	17.27 ms	0.19 ms	0.79 ms	-0.22 ms

レジスタ	0	1	2	3	4	5	6	7	8	20	HRL
データ	89	E	1D	83	237	5	1E	23E	1B	16	0

推奨モニタ）CZ-6xxD,CU21他

備考）以前のCRT960掲載時に同時に載せたもの
多少広いだけだが，垂直が55Hzあるので見やすい

5）表示ドット）横 1023　縦 762　　インタレース
ドットクロックオシレータ＝69.551MHz　　8ドットクロック＝0.23μsec

	同期期間	データ表示期間	同期パルス幅	バックポーチ	フロントポーチ
水平	32.90μs/30.4kHz	29.45 μs	0.69 μs	2.07 μs	0.69 μs
垂直	12.99ms/76.96Hz	12.57 ms	0.07 ms	0.30 ms	0.07 ms

レジスタ	0	1	2	3	4	5	6	7	8	20	HRL
データ	8E	2	7	87	18A	1	A	188	1B	1A	0

推奨モニタ）CR-7000,F557,MF-8617他

備考）本文中で求めた，1024x768モード，インタレース
垂直同期が高いので短残光モニタでもそれほどちらつかない
私としては2を使うよりもお勧め
CR7000ではほぼ完璧といえるだろう（多少H-CENTの調整が苦しいが）

6）表示ドット）横 1023　縦 798　　インタレース
ドットクロックオシレータ＝69.551MHz　　8ドットクロック＝0.23μsec

	同期期間	データ表示期間	同期パルス幅	バックポーチ	フロントポーチ
水平	32.90μs/30.4kHz	29.45 μs	0.69 μs	2.07 μs	0.69 μs
垂直	13.55ms/73.78Hz	13.16 ms	0.07 ms	0.30 ms	0.03 ms

レジスタ	0	1	2	3	4	5	6	7	8	20	HRL
データ	8E	2	7	87	19B	1	A	19A	1B	1A	0

推奨モニタ）CR-7000,F557,MF-8617他

備考）5で垂直同期のいけにえにして，縦の表示を多くしたもの

7）表示ドット）横 1023　縦 1024　　インタレース
ドットクロックオシレータ＝69.551MHz　8ドットクロック＝0.23μsec

	同期期間	データ表示期間	同期パルス幅	バックポーチ	フロントポーチ
水平	32.90μs/30.4kHz	29.45 μs	0.69 μs	2.07 μs	0.69 μs
垂直	17.24ms/58.01Hz	16.84 ms	0.07 ms	0.30 ms	0.03 ms

レジスタ	0	1	2	3	4	5	6	7	8	20	HRL
データ	8E	2	7	87	20B	1	A	20A	1B	1A	0

推奨モニタ）CR-7000他

備考）CR-7000で使える最大サイズの画面モード
ただしちらつく

8）表示ドット）横 1023　縦 766　　インタレース
ドットクロックオシレータ＝69.551MHz　8ドットクロック＝0.23μsec

	同期期間	データ表示期間	同期パルス幅	バックポーチ	フロントポーチ
水平	37.96μs/26.3kHz	29.45 μs	2.30 μs	4.83 μs	1.38 μs
垂直	15.79ms/63.33Hz	14.58 ms	0.11 ms	0.99 ms	0.11 ms

レジスタ	0	1	2	3	4	5	6	7	8	20	HRL
データ	A4	9	1A	9A	19F	2	1C	19C	1B	1A	0

推奨モニタ）CU21他

備考）1024×848モードの縦ラインを少なくし，CU21の垂直同期ぎりぎりまで，上げたモード
垂直ブランキング時間を多少ケチれば，800ぐらいまでいくはず
編集室のCU21で使用しているモード。でもちらつく

9）表示ドット）横 1024　縦 1022　　インタレース
ドットクロックオシレータ＝69.551MHz　8ドットクロック＝0.23μsec

	同期期間	データ表示期間	同期パルス幅	バックポーチ	フロントポーチ
水平	32.21μs/31.0kHz	29.45 μs	0.46 μs	1.84 μs	0.46 μs
垂直	16.97ms/58.92Hz	16.49 ms	0.10 ms	0.29 ms	0.10 ms

レジスタ	0	1	2	3	4	5	6	7	8	20	HRL
データ	8B	1	5	85	20E	2	B	20B	1B	1A	0

推奨モニタ）発見できず

備考）CZ-6xxD，CU21で追従できたが，左右がはみ出たり，画面が多少腐ったりする
そのままでは絶対使えないが，なぜ，掲載するかは各自考えること

10）表示ドット）横 1023　縦 1022　　インタレース　要改造
ドットクロックオシレータ＝80.000MHz　8ドットクロック＝0.20μsec

	同期期間	データ表示期間	同期パルス幅	バックポーチ	フロントポーチ
水平	28.20μs/35.5kHz	25.60 μs	0.20 μs	2.00 μs	0.40 μs
垂直	14.83ms/67.42Hz	14.44 ms	0.03 ms	0.31 ms	0.06 ms

レジスタ	0	1	2	3	4	5	6	7	8	20	HRL
データ	8C	0	6	86	20D	0	B	20B	1B	1A	0

推奨モニタ）CR-7000

備考）ドットクロックオシレータを80MHzに換えたときにできる1024×1024ドット画面
かなりおすすめ

11）表示ドット）横 680　縦 767　　ノンインタレース
ドットクロックオシレータ＝100.000MHz　8ドットクロック＝0.16μsec

	同期期間	データ表示期間	同期パルス幅	バックポーチ	フロントポーチ
水平	17.60μs/56.8kHz	13.60 μs	1.76 μs	1.92 μs	0.32 μs
垂直	14.19ms/70.49Hz	13.52 ms	0.11 ms	0.51 ms	0.05 ms

レジスタ	0	1	2	3	4	5	6	7	8	20	HRL
データ	6D	A	12	67	325	5	22	322	1B	16	0

推奨モニタ）F557,MF8617

備考）ドットクロックオシレータを100MHzに取り換えて，私が最初に作ったモード
512×512ドットとほぼ等しいアスペクト比で表示範囲が広くなる

12）表示ドット）横 1024　縦 767　　ノンインタレース　要改造
ドットクロックオシレータ＝100.000MHz　8ドットクロック＝0.16μsec

	同期期間	データ表示期間	同期パルス幅	バックポーチ	フロントポーチ
水平	24.32μs/41.1kHz	20.48 μs	1.12 μs	2.24 μs	0.48 μs
垂直	19.26ms/51.92Hz	18.68 ms	0.05 ms	0.49 ms	0.05 ms

レジスタ	0	1	2	3	4	5	6	7	8	20	HRL
データ	97	6	10	90	317	1	15	315	1B	16	0

推奨モニタ）F557,MF8617

備考）目玉商品。1024×768のノンインタレース。垂直同期が少なめだが，かなり使える
私がよく使うモード。ドットクロックをもう少し上げられればなぁ……

13）表示ドット）横 999　縦 898　　インタレース
ドットクロックオシレータ＝80.000MHz　8ドットクロック＝0.20μsec

	同期期間	データ表示期間	同期パルス幅	バックポーチ	フロントポーチ
水平	27.60μs/36.2kHz	25.00 μs	0.20 μs	2.00 μs	0.40 μs
垂直	12.81ms/78.09Hz	12.42 ms	0.03 ms	0.30 ms	0.06 ms

レジスタ	0	1	2	3	4	5	6	7	8	20	HRL
データ	89	0	6	83	1CF	0	B	1CD	1B	1A	0

推奨モニタ）F557,MF8617,CR7000

備考）水平を1000に減らして，垂直を900に減らしたため，垂直同期にずいぶん余裕ができた
ドットクロックオシレータが69MHzでも使えると思われる
ちらつきはほとんどない

14）表示ドット）横 1023　縦 1024　　インタレース
ドットクロックオシレータ＝100.000MHz　8ドットクロック＝0.16μsec

	同期期間	データ表示期間	同期パルス幅	バックポーチ	フロントポーチ
水平	24.16μs/41.4kHz	20.48 μs	0.96 μs	2.24 μs	0.48 μs
垂直	12.78ms/78.24Hz	12.37 ms	0.05 ms	0.31 ms	0.05 ms

レジスタ	0	1	2	3	4	5	6	7	8	20	HRL
データ	96	5	F	8F	210	1	E	20E	1B	1A	0

推奨モニタ）F557, MF8617

備考）ドットクロックオシレータが100MHzだからできた最強モード
実画面全てを表示し，かつ，ちらつきはほとんどない
私が普段，T660iJで利用しているモード
なまじノンインタの12番よりも目はつかれないかもしれない

15）表示ドット）横 1000　縦 898　　インタレース
ドットクロックオシレータ＝69.551MHz　8ドットクロック＝0.23μsec

	同期期間	データ表示期間	同期パルス幅	バックポーチ	フロントポーチ
水平	31.98μs/31.3kHz	28.76 μs	0.46 μs	2.07 μs	0.69 μs
垂直	14.87ms/67.25Hz	14.39 ms	0.10 ms	0.29 ms	0.10 ms

レジスタ	0	1	2	3	4	5	6	7	8	20	HRL
データ	8A	1	6	83	1D0	2	B	1CD	1B	1A	0

推奨モニタ）CR-7000

備考）横ドットを1000に減らして，縦を900に設定してみた

16）表示ドット）横：1023　縦：768　　インタレース
ドットクロックオシレータ＝69.551MHz　8ドットクロック＝0.23μsec

	同期期間	データ表示期間	同期パルス幅	バックポーチ	フロントポーチ
水平	37.96μs/26.3kHz	29.45 μs	1.84 μs	5.29 μs	1.38 μs
垂直	15.79ms/63.33Hz	14.35 ms	0.04 ms	1.40 ms	0.00 ms

レジスタ	0	1	2	3	4	5	6	7	8	20	HRL
データ	A4	7	1A	9A	19F	0	25	19F	1B	1A	0

推奨モニタ）CZ-6xx

備考）標準モニタでできる究極のモード？　かなり厳しい設定なのでモニタによっては
追従しない可能性が高い
それなりにちらつくのでOAグラスなどの使用が望ましい
CU21系はおそらく不可

[特集]X68000と仲間たち

手軽さ・軽さ・大きさで見る

当代ハードディスク事情

Ishibumi Akira 伊溢見 あきら

外部記憶装置ハードディスク。もはや私たちのパソコンライフに欠かせないものになってきています。ここでは大きなものから小さなものまで，ハードディスク導入の指針を見ていきましょう。

数年前ならわかりませんが，いまどきハードディスクをまったく使わずにコンピュータを100%活用するなんてことは，北極海で吹雪のなかシンクロナイズドスイミングをするようなものです。危険というのではなく，それは限りなく無謀であるといえます。そこで今回は，いまどきのハードディスクライフスタイルを検証し，目的を持ったハードディスクの有用性を説いていきたいと思っています。

我使う，ゆえにそこにあり

ハードディスクをなにに使うかと聞かれて最初に思いつくことはOS環境の構築です。というか，ハードディスクを新たに使い始めるためにはOS環境を変えなければならないので，ハードディスクとOS環境は一体のものであると考えてもよいでしょう。

本体に標準装備されているフロッピーディスクというメディアは，X68000の処理速度を考えると決して十分に速いメディアとはいえません。フロッピーディスクとはその可搬性とわずかばかりの経済性によってその価値を維持している，いわば過去のメディアです。

これを使って，多様化する一方の用途，プログラム，CG(画像処理)，音楽などに対応しようとすれば，まず第一に速度の問題にぶつかり，そしてすぐに第二の容量の問題が現れ，フロッピーディスクの限界というものを簡単に体験することができます。

ここで少なくともハッキリしていることは，ハードディスクはフロッピーよりも十分大きく，そして十分速いという事実です。基本的なOS環境にハードディスクは必ずしも必要なわけではありませんが，そこに根差す多くの有用で素晴らしいアプリケーションのために，ハードディスクは求められているのです。

さらに最近ではSX-WINDOWの普及もめざましく，これに依存したマシンオペレーションを行うとすれば，より外部記憶の速度と容量が求められていきます。ちょっと凝ってフォントを入れるようになると，もうどれだけ容量があればいいのか簡単に答えが出るものではありません。PC-9801やAT互換機におけるWINDOWSでも，あればあるだけよいとされています。

GUIはオペレーション情報を視覚的に補填する分，その分の容量を求め，付随するデータによって処理が遅れることを防ぐために速度を要求します。これは，本体にとっても外部記憶装置にも共通の傾向だと考えなくてはいけません。こうした高度化されたOS・アプリケーション環境下で扱うと，データも同調して増加していくことは当たり前の帰結であるといえます。

むしろ，そのサイズの大小を論じるのではなく，サイズを考えないで済むといい換えたほうがいいでしょう。

物理的な大きさを考慮しないでファイルを取り扱えるようにするには，またここにも尨大な記憶領域が必要になってきます。それを管理するならば外部記憶との転送がデータのサイズによらない速度である必要があります。つまりデータに対しても高速で大容量の現実的なメディア，つまりハードディスクが必要になっているのです。

ハードディスクの環境は当初，贅沢品であったものが，いまやなんでもなく，視力の悪い人が眼鏡やコンタクトレンズを使うような感覚で，すでに当たり前になってしまっているものなのです。

容量と速度の歴史

X68000シリーズは当初からハードディスクインタフェイスを内蔵していました。しかし純正のハードディスクは10Mバイト348,000円という値段でしたから，まだまだ一般には普及していきませんでした。

最初に大きく現れたハードディスク環境というのは，ACE-HDの20Mバイトという世界でした。これは明らかにOS環境だけをターゲットにしていたものです。この頃はだいたいハードディスクの単価が1Mバイトあたり5,000円でしたから，まだまだ高嶺の花の部類であったということができるでしょう。

X68000のインタフェイスは独自のものですが，9801-27互換(NECの純正インタフェイスの型番からこう呼ぶのが一般的)と呼ばれるSASIに似たものを採用しており，外付けならばPC-9801のハードディスクがつながるという，意外と互換性の高そうな形式のものでした。

ただ，この頃のPC-9801用ハードディスクには純正ボード以外にも，メーカーごとにさまざまなローカル規格がありました。それは，非純正のボードの速度によるもので，少しでも速いアクセスをセールスポイントにするため，多くのローカル規格が大手を振って歩いていたのです。この頃だとアクセス時間が30m秒を切る程度で速いといわれていました。あくまでも比較の対象はフロッピーだったからですが，結局はかなり遅い代物だったといってもいいでしょう。

ローカル規格のおかげで，使えるのかどうかを地道に確認する必要があり，X68000で使えるメーカー保証の情報も乏しいなかでは価格を考えるとかなり導入に難関が多かったことは確かです。ACE-HD，EXPERT-HDなどのハードディスク内蔵モデルにかなり人気が集まりました。

C言語の開発環境にとっては，ハードディスクは必需品となってきたこともあり普及段階に入ってきた時期だといえます。

しかし，小さな容量のハードディスクでは使っているうちに極端に速度が落ちてきましたし，いまも現役のEXPERTやPROにハードディスクをつけようと探している人は，もっと大変な状況になっているので，

振り返るとこれらのマシンは，あまりハードディスクの恩恵を受けたマシンとは呼び難いと考えることもできます。

結局，X68000に本格的なハードディスクの時代が来るのは，SCSIを標準装備したSUPER以降ということになるでしょう。そして，それは同時に事実上SASIのハードディスクが末期を迎えつつあったということになります。だいたいその頃で，1Mバイトあたりが2,000円くらいでしたから，商品の主流は40Mバイトから80Mバイトといったものでした。モノの大小は相対的なものですから，当時は80Mバイトでも大容量と呼ばれていたものです。このあたりのクラスでアクセス時間は20～25m秒と，速度も次第に上がってきつつありました。

1Mバイトあたり2,000円の壁を割る頃になると，40Mバイトが10万以下で手に入るようになったためか，ハードディスクは，大きな波を迎えます。相変わらず，SASIに関してはあまり選択の面で芳しくはありませんでしたが，それでもこの頃に相当多くの人がハードディスクの世界に足を踏み入れたはずです。

需要の増加に連鎖して起こるのは大量生産とそれによるコストダウンでした。それから1年ほどたってXVIがデビューする頃には，1Mバイトあたりが1,000円になり，80Mバイトですら10万円を切るというところまでコストが下がっていったのです。そして多くのメーカーはアクセス速度20m秒以下で，キャッシュメモリを搭載したSCSI機を主力製品とし，容量をより増加させ，より低価格のものを競って提供するようになります。ここでハードディスクは，特別なものではなくなり，標準ともいうべき，当たり前の周辺機器に変わったのです。

それから先は，倍々ゲームでコストが下がっていき，ハードディスクの価格はその依存度と反比例して，どんどん下がってきています。いまではものによっては1Mバイトあたり100円程度のものもあり，経済性でもきわめて優位なメディアとして確立したのは疑いようのない事実でしょう。

こうした，機種が豊富で大容量で高速で低価格な時代におけるハードディスクの選び方や使い方を問われる時代がやってきたのです。

正しい選択のために

まず新たにハードディスクを導入するためには，自分がどんなシステムを持っているのか把握しなくてはいけません。一般的に考えて状況は，以下に挙げる5とおりのいずれかになるはずです。

1：EXPERT2以前(SASI)内蔵なし
2：EXPERT2以前(SASI)内蔵あり
3：EXPERT2以前SCSIボード装備
4：SUPER以降内蔵なし
5：SUPER以降内蔵あり

ここで，1と2に関しては，もう選択の余地はなく9801-27互換でつながるSASIのハードディスクを探す以外に道はありません。モノがそもそもないとか，不当に高いとかというのは承知のうえで，選択するしかありませんから，それ相応の覚悟が必要です。また，ドライブはせいぜい80Mバイトが最大で，なおかつそれを40M×2としてしか取り扱えない，IDが0と1しか設定できないドライブは内蔵型と同居できない，ドライブ自体が遅いといった難問も一緒に抱え込むことになってしまいます。個人的にも現実的にもあまりすすめられるものではありません。

すでにスロットが満杯の人には申し訳ありませんが，ここはぜひSCSIボードを導入し，選択の幅を大きく広げていただきたいと思います。ショップを見て回って現実を認識しなくても，これがもっとも正しい選択だといい切ってもいいでしょう。

ここであえて書いておきますが，通信などで出回っている，SASI搭載機種にSCSIのハードディスクをつなぐフリーソフトウェアのドライバの機能に期待して，SCSIのハードディスクを買うようなことは避けたほうがいいでしょう。

実際にどうなるかは，やってみないとわかりませんし，入手方法も商業ベースの一般的なものではありません。もちろん接続レポートがあってもそれが保証になるわけではありませんので，なんらかの知識や確証，失敗したときの補償でもない限り，あえて危険に挑む必要などないでしょう。

安くなったとはいえ，それは容量との価格比であり，単体の商品としてのおおまかな価格帯はそれほど変化していませんから，同じ金を払うならば，安全や確実のために使うほうが手堅い選択だといえます。

SCSIで決めてみる

それでは3，4，5のケースにあたる，最も現実的なSCSIによる接続を考えてみましょう。SCSIについては，SCSIと書いてあればなんでもつながるように思われていますが，実際には接続するハードディスクが512バイト/セクタでないと使用することはできません。X68000対応と明記していないものは所詮無保証ですが，一応これやMacintosh，FM TOWNS対応などという言葉はアテにできるようです。

選択の余地が多いSCSIハードディスクですが，あまりにその数が多いため，なんらかのテーマを持ってハードディスクを選ばないと，あっちもいい，こっちもいいというように目移りし，結局どれが自分にあっているのかわからなくなってしまいます。そこで今回は次のような選択テーマを用意して，ハードディスクの使用スタイルを考えてみることにしました。

A：手軽さ

家庭の事情その1

当家のハードディスクは，一世を風靡した小容量の240Mバイト(実際には234Mバイトとさらに少ない)です。最近は「小容量」というのを強調しても怒られないようになりましたが，1年くらい前から狭さには苦労しっぱなしです。周囲には，MOがあるではないかといわれたりもするのですが，あれはフロッピーの代わりでしかありませんし，逆にMOのために余計窮屈になってしまっている部分があるのが現状です。

まずAドライブはシステムです。これの容量が13Mバイトです。主に実行ファイルだけで，アプリケーションなどのセットは，ここにはありません。Human68kのオリジナルファイルは¥DOS¥にひととおり入れてあり，その他のファイルは¥BIN¥に分類しています。辞書はRAMディスクなので専用パーティションはありません。

次がBドライブで，ここは大型アプリケーションのインストール領域です。大型アプリケーションにはゲームというもの(？)や，BASIC(！)も含まれていますので，40Mバイトを配分してあります。部屋では，紙に書くと行方不明になるので，メモの類もエディタで打ち込んでここに入れています。

Cドライブは通信のデータ待避領域になっており，これが60Mバイト確保してあります。とはいえ，最近は巨大なデータが多く，この容量でもこまめに整理しないと，ヤバイことになってしまうのが難です。

最後のDドライブはMOの編集用のバッファで，MO1枚とまったく同じ容量の121Mバイトが切ってあります。容量が同じだとdiskcopyで転送できて便利なのですが，これだけでは増え続けていくMOを整理するのには力不足というしかありません。

こういうことを考えると，240Mバイトというのは，そろそろ限界になってきつつあります。不足する容量を補うつもりで導入したMOのために，さらにハードディスクの容量が必要になるというのは予想外でした。とりあえず，次の目標は1.6Gバイトあたりかなぁ。

(伊藻見あきら)

B: 軽さ

C: 大きさ

まずAの手軽さというのは，気楽に買える価格帯で，なおかつOS環境やデータ保存には，十分な容量を持っているというタイプを指します。いままでハードディスクのなかった環境の場合は，これがもっとも無難な選択でしょう。ただドライブ本体はティッシュペーパーの箱（少なくともその半分）くらいの大きさがあり，それなりに場所を食います。しかし，現在もっとも売れている価格帯のせいか，安定した商品が多く安心して買うことができるというメリットがあります。機種が多いですから，性能だけでなくデザインやカラーリングの面を考えに入れてもよさそうです。特に電源スイッチが正面についているようなタイプは，X68000では重宝するでしょう。

これらのタイプに分類される機種は，1Mバイトあたり200円程度のもので，下は200Mバイトから上は500Mバイト程度までの容量のバリエーションが一般的です。本誌の広告にも多く見られるので，価格を確認してみるのもよいかと思います。

こういった手軽さを特徴に持った商品のなかには，機能的にいくつかの制限がついているものもありますので注意が必要です。特にそのなかでも，ID番号固定になっているタイプは要注意で，2台連結してデータのコピーができなかったり，内蔵とバッティングして内蔵側のIDを変更したらドライブの順番が狂ってしまったり，といったようなことが起こりえます。一時的なトラブルかもしれませんが，ややこしいことがわかっているならそれは避けるべきだといえるでしょう。

そういったことに気をつける必要はありますが，2年ほど前に大容量といわれていた機種が半分以下の値段で手に入るわけですから，ちょっと無理してでも，結局はお買得であるという印象を受けます。最初の1台に迷っている人は，このあたりの機種を選ぶといいでしょう。

ただ，いくらお買得だからといっても，現在ハードディスクをすでに使っている人，特に外付けのものを使っている人が，容量アップのための増設機として選ぶことについてはあまりメリットがあるとは思えません。前述のIDのバッティングなどもそうですが，やはり同じようなものをいくつも置くようだと場所も取りますし，コンセントを余計に取るわ，熱は出すわ，ファンが合唱するわ，スイッチをそれぞれオン-オフするのが面倒だわと，あまり賢い選択のようには思えません。

またドライブが物理的に別れているので，パーティションの切り方にも制限が生じてしまいます。でも，切実に容量不足を痛感している人にとっては，避けて通れない問題でもあります。かといって，いまのものを手放して買い替えるというのも，ちょっと勇気がいります。そこで逆に2台あるというメリットを活かせるようなハードディスクの増設を考えてみることにしましょう。

小さくても大きくてもOK

ここで出てくるテーマは，Bの軽さです。つまり，なるたけ場所を取らないような，邪魔にならないハードディスクの導入を目指そうということになります。

昨今の技術進歩で，いままでとは比べものにならないくらい小型のハードディスクドライブも開発されており，実際の製品にもそういった超小型機が現れてきています。価格面では，テーマAのものとはやや差がありますが，カセットテープのケースを2，3個張り合わせたようなその小ささを見ると，見た目以上のメリットを感じる人もいることは想像に難くありません。

容量のバリエーションは，だいたい最近流行の100M～300Mバイトとなっており，単体でも十分な容量をもっています。価格的には1Mバイトあたりが300～400円くらいとやや高価になっていますが，それだけの価値のあるものだといえます。

もちろん最初の1台目にこれを選んでも何も問題ありません。本体のツインタワーの上にちょこんと置いて使えば，特別の設置面積が不要の，限りなく内蔵に近い感覚で使うことができます。しかしなによりも，簡単に持って移動することができるようなサイズであるだけに，自分の環境をあらゆるところに持ち運ぶことが可能になるというメリットも見逃せません。

ただし，こういった持ち運びでの使用は，やはり常に自分のシステムに常駐するハードディスクが別にないと，万が一の場合に不安になります。それがゆえに増設向きの導入スタイルということができます。

データなどを持ち歩いたりするのにも好都合ですし，MOと違って相手先にドライブがつながっていなくてもファイルの交換が可能です。ちょっと気になるのは，あまりに小さいためドライブにコネクタがひとつしかなく，SCSIのデイジーチェーンの途中に挟めないようなものがあるという点です。この場合，そのユニットは末端専用になってしまうので，固定IDのバッティングのように2台を並べて使うわけにはいかなくなることが考えられます。この点にだけは注意が必要でしょう。

意外な機動性と便利さを手に入れつつ，容量不足という問題を解消できるといった意味では，これはなかなか賢い選択といえるかもしれません。

しかしなかには，100Mバイトに200Mバイトなどを足しても，その場しのぎの間に合わせにしかならないことを薄々感じている人もいるかと思います。もったいなくて捨てられない，決して減ることのないデータやソフトの山は多くなればなるほど一括した管理を行わないとカオスの世界へまっしぐらです。必要なもの，そうでないものを見分けることすら困難になってしまう前になにか手を打つ必要があるのです。

そこで，最後のテーマである，Cの大きさをハードディスクに求めることになるのです。大きいというのはいままで出てきたような，M(メガ)程度の可愛いサイズではありません。今の時代の大容量とは，ずばりG(ギガ)クラスのハードディスクということになります。

さすがに，このクラスではカセットサイズというわけにはいきませんが，基本的にはティッシュボックスとそれほど変わらない大きさを保っています。それに容量単位のコストはテーマAで挙げたようなものと，同等もしくは若干下回るくらいです。お金の許す限りの大容量で将来に備えたいような人は，考えに入れてみるといいでしょう。

それどころか，ドライブメーカーから見た主流というのは，このクラスに移りつつあるらしく，アクセス速度が10m秒を切るものは珍しくありません。性能的にも価格的にも，非常に大きな可能性を秘めているといえます。

MOドライブユーザーにとって課題である，複数メディアを対象としたファイル整理なども，これなら問題ない容量のパーティションを切ることもできます。

さらに，一部のショップにおいてはドライブを選んで，自分でケース(一緒に売っている)に組み込むような，まるでハードディスクキットのような販売をしているところがあります。そういうところでのメインもやはり，このGクラスのものが多く，メーカー製完成品のものと比較すると，その価格は驚くほど差があります。

例を挙げてみるとケース込みの1.6Gバイトで16万円，1Mバイトあたり100円という驚異的なコストパフォーマンスのものが

できあがってしまうのだから驚きです。ただ，いくら安いといっても，容量も巨大ですからそれなりの金額になることには目をつぶるしかありません。

また，こういったキット形式では，やはり機械工作の苦手な人は組み立てのできる人に任せたほうが安全です。そういう環境に恵まれない人には，こういったものには手を出さないような勇気も必要でしょう。

実は，さらに上には上があるらしく，個人輸入でアメリカなどからドライブを買ったりすれば，円高などの要素も絡んで，さらにお得なハードディスクを手に入れることも可能なようです。ただ，ここまでくると語学力や，リスクといった別件の要素が入ってくるので，安くなる分，安心や安全から離れていってしまいます。境目は非常に難しいですが，先にも書いたように，金で買える安全は，買ったほうがいいと思いますから，この方法はあまりおすすめしません。

手に入れてはみたけれど

こうして多くの商品から自分にあった商品をみつけて購入が終わると，今度はそれを使うという，もっと大変で難しい作業が待っています。まあ，最初にケーブルをつなぐときはハーフサイズかフルサイズかを確認してあれば問題ないし，IDも重ならないようにしてやれば番号がトビトビでも問題なしですから，問題は別のところにあります。普通はとりあえずつなげば使えるようになりますが，まれに問題が発生することがあります。

簡単なものには物理的な接続の順番，ターミネータの有無といった部分でのユニットの不安定性のようなものがありますが，これに対する明確な対処法というものは，正直なところありません。行き当たりばったりで，いろいろと試して安定する組み合わせを探すしかないようです。経験者になって懲りてくると，だいたい予想がついてしまうようですが，あまりトラブルに慣れたいとは思いませんね。

いちばん深刻なものには，ローレベルフォーマットの問題があります。ローレベルフォーマットというのは，ハードディスクの物理フォーマットをするもので，これをしないと領域確保などの，各種管理をすることができませんから，そのハードディスクが使用できないということになります。しかしHuman68kのformat.xには，明示的にローレベルフォーマットを指示する項目がありません。また古いバージョンでは，どうやらローレベルフォーマットができないようです。そういった意味で，キット形式で組み上げたハードディスクなどの場合，いったん別な機種でローレベルフォーマットしたり，フリーソフトウェアのフォーマットプログラムを入手するなどの処置を取る必要があるかもしれません。

なんとかローレベルフォーマットできてしまえばそれでいいかというと，体験からいわせてもらうと，そうでもありません。一部のソフトでローレベルフォーマットを行った場合，領域確保時のクラスタ数が同じ容量を指定しているにもかかわらず一致せず，diskcopyコマンドが使用できなくなり，非常に困ったことがありました。

こういった可能性がなきにしもあらずですから，使用にあたっては，できるだけ同一の環境を確保してやる必要があります。

誰がためのハードディスク

領域確保の場合にも気をつける必要があります。ファイル管理の関係上，64Mバイト以下のパーティションは1クラスタ1Kバイトですが，それ以上は1クラスタ2Kバイトになり，さらに128Mバイトを超えると1クラスタ4Kバイトになってしまい，ディレクトリひとつエントリーするだけでも，それだけの容量が消費されてしまいます。

ファイルサイズの切り上げ単位が大きくなることでその切り上げ分の無駄，俗にいうクラスタギャップが大きくなるので，用途にあわせたパーティションサイズを用意してやるとよいでしょう。

数百Mバイトを超えてしまうと，計画的で論理的なハードディスクの管理は不可欠の要素となってきます。小さい入れものは，なにが入っているかすぐにわかりますが，巨大なバケツの中にあるものを探すのは容易なことではありません。

技術の進歩で身近になった，数年前からは想像もつかないハードディスク空間は，使う側の私たちの意識をも進歩させないと使いこなすことはできないでしょう。ハードディスクはコンピュータのものですが，コンピュータが人のために作られている以上，それが役に立つかどうかはそれを使う人にすべてがかかっているのです。自らが手に入れたデータに溺れないためにも，どうかそのことを忘れないでください。

小さいことは正義である・HDD編

緑電子から発売されているV2-200(通称NOVA2)は，2.5インチドライブを採用した200MバイトSCSIハードディスクです。

この製品のウリは「小さいこと」これに尽きます。本体の寸法は45×158×90㎜となっており，これは単行本を3冊重ねた大きさよりも，さらにひと回り小さい程度です。さらには本体重量650グラムと軽量ですから，なんの苦労もなく持ち運ぶことができます。据え置きで使うにはもちろん，移動用としてこのサイズは魅力的だといえるでしょう。

「自分の環境を常に持ち歩く」そんな用途に最も適しているものというと，現在ではMOが第一に挙げられますが，これはドライブが移動先にないと話になりません。しかしハードディスクならそんな心配も無用です。もちろんSCSIマシンがなくてはいけませんが。

性能についてはキャッシュがついていないことがちょっと不足なくらいで，あとはごく普通のハードディスクです。転送レート上限が10Mバイト/秒のドライブを使用しているので，普通のX68000で使う分には速度がネックになることはまずありません(XVIで600～700Kバイト/秒程度)。高速化したXVIではシーケンシャルリードで約900Kバイト/秒という値が出ています。

シーク速度も平均12msとそこそこ高速ですから，「ディスクキャッシュは怖くて使えない」という人もストレスが溜まることはないでしょう。

気にする人は気にする動作音ですが，これはきわめて小さく，アクセス時以外はほとんど無音です。電源を投入してから動作可能な状態になるまでの待ち時間も短く，容量や速度なども含め，トータルな使用感はなかなか良好です。

デザイン的にも，どうも「ダサい」周辺機器が多いなか，なかなか健闘しているといえますし，ボディの配色はチタンブラック系統の暗いグレーですから，X68000と並べても浮いた存在になることはないでしょう。

同容量の3.5インチハードディスクより少々割高ですが，小さいことによるさまざまなメリットは値段には代えがたいものがあります。1台目，増設用を問わずおすすめの製品です(NOVA2シリーズには，さらに大容量のタイプも用意されています)。

PC-9801用として販売されているものでも，問題なくX68000に接続できますから，こちらの方面から安い店を探してみてもいいでしょう。

(進藤慶到)

安全さを考えたインストール例

ちょっと大きめの常用HD

Tomonaga Takeaki 友永 健明

中規模HDただしちょっと大き目，という500MバイトクラスのHDです。ここでは初心者向けに常用HDシステムを組むノウハウを見てみましょう。大き目だからこその使い方というものもあります。

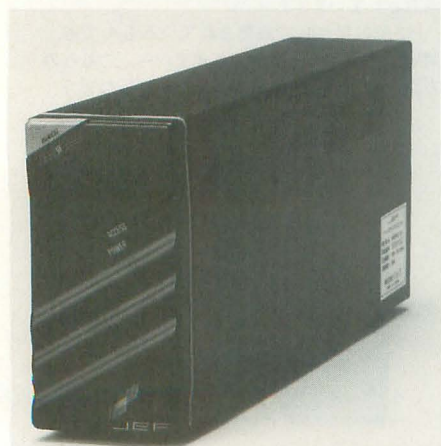
ジェフの新しいハードディスクを試用してみました。500Mバイトクラスですから，私が普段使っているドライブよりはずいぶん大きくなります。このクラスがいまや「ちょっと大きめ」という感じなのですから，時代の流れは速いものです。

このユニットは，かなり標準的に使われているQuantum製のドライブを使用しており，非常に高速です。通常のコマンドを起動するだけならほとんどアクセスを感じさせません。フォーマット後の総容量は516Mバイトになります。

安全なシステムを作る

新しいハードディスクとなれば，やはり自分にあった環境のインストールから始まります。こういった，容量にかなり余裕があるハードディスクの場合，なにをどう使ったらいいのか迷ってしまう人もいるかもしれません。また，インストール作業を必要以上に難しいものだと思っている人もいるようです。

まずはどのようにパーティションを切るかというところが問題になってきます。

私はいろいろ苦い経験も積んでいますので安全性に重点を置き，

A: システムドライブ 2Mバイト
B: 辞書ドライブ 5Mバイト
C: ツールドライブ 121Mバイト
D: データドライブ 121Mバイト
E: ワークドライブ 121Mバイト
F: ワークドライブ2 155Mバイト

という構成を考えてみました。

要所を121Mバイトで確保しておけばMOとのあいだでDISKCOPYコマンドによる高速転送ができます（とはいっても，かなり時間はかかる）。いまMOを持っている人はもちろん，まだ持っていない人も備えておいて損はありません。

ドライブAはわずか2Mバイトです。これは，「システムのバージョンが0.1上がった」などというときに対処しやすいようにと編み出された生活の知恵です。

OS本体が前のバージョンよりちょっとでも大きくなると差し換えが大変になるので，ドライブ丸ごと交換できるようにOS以外は入れないことにしているわけです。Human68kとCOMMAND.Xだけでもよいのですが，システムディスクの内容はすべて入れてみました。

ちなみに，今回は十分にワークドライブを設定しているので，このような方法をとらなくても起動ドライブの退避ができるのですが，やがてくると思われる「なぜか空きがない」状態を考慮して万全の備えをしてみました。まあ，システムの入れ換えはバックアップのついでにすればいいのですが……。

Bドライブは辞書用です。しかし，Aドライブが飛んだ場合に応急処置できるようにBドライブにもシステムを入れておきます。最低限，立ち上がってメンテナンスができるようなツールを入れておきます。それが終わったら辞書を入れます。

なお，ヘルプキーを押しながら起動するとどのパーティションから立ち上げるか選択できますので，起動チェックもしておきましょう。

Cドライブには常用ツールを入れます。SX-WINDOW関係，C言語関係，グラフィックツール関係，音楽ツール関係，その他各種ツール関係……。121Mバイトあれば余裕はありますが，それでも結構埋まるものです。

ツールだけでなく，システムに必須で，かつあまり更新しないデータ，すなわちアウトラインフォントやAD PCMデータはここに入れておきます。

Dドライブにはテキストデータ，常用のグラフィックデータ，音楽データ，アニメーションデータなどが入ります。「常用の」ということは常用でないものもあるわけで，それはMOにまとめてあります。

ちなみにバッチファイルなどはAドライブに入れます。ここは1セクタ1Kバイトですから（貧乏性！）。

最後にMOを整理することを考えてワークドライブを設定します。

このような構成であれば，重点的にバックアップをとるのはCドライブだけで済みます。

*　　*　　*

と，まあ，こんな具合です。

しかし，最近のハードディスクは安くて速いし，なにも考えなくてもつながるから楽になりましたねぇ。

こだわらない人には関係ありませんが，最近は黒いハードディスクが少なくなっていますからこういう製品は貴重でしょう。

難点は電源スイッチが本体裏面の左下隅（SCSIコネクタの下）にあるということでしょう。私はほとんどパソコンの電源を落とさない人なのでいいのですが，普通の人は絶対困るはずです。もっとも手が入りにくいところに電源スイッチがあるのですから。やっぱ，連動コンセントでも作りますか？

GF-540q　128,000円（税別）

選択から基本操作まで

MOドライブの一般的使い方

Nakano Shuichi 中野 修一

すっかり一般的なメディアとなった感のある3.5インチ光磁気ディスク。フロッピーディスクとハードディスクの長所を持ち合わせています。ここでは導入の手引きから，一般的な使い方までまとめて解説してみましょう。

普及価格になったおかげで光磁気ディスクもずいぶん浸透してきました。

どれくらいユーザーが増えたかなと，3月号で行った愛読者アンケートの束のなかから適当に100枚分数えたところ，所有者15人……。これに答えてくるのはヘビーユーザーが多いでしょうから，ちょっと例外として，改めてはがきを集計すると所有率8%になりました。1993年9月号の特集時にはがきから集計したものが3%でしたから，9カ月で3倍近くに所有者が増えたことになります。

MOドライブを選ぶ

まずは適当なドライブを選ばなければなりません。

●X68030の場合

特になにも問題はありません。

●X68000の場合

X68000用として売られているものが無難です。Human68k ver.3.0を使ってもSCSIのINQコードが変更できないドライブでは使用できません。確認が必要です。

INQコードが変更できないものはX68000のSCSIボードでは対応していない機器ですので，そのままでは使用できません。これに対応したフリーソフトウェア，INQPATCHかSxSIを使うことになります。

SxSIはX68000でMS-DOSマシンのハードディスクをアクセスするために作られたドライバです。付属機能が多く，そのうちのいくつかは非常に有用です。詳しくは紀尾井氏の記事をご覧ください。

なにを選ぶか

では具体的にどのような製品を選ぶべきかを考えてみましょう。

まず重要なのは内部ドライブのメーカーを調べることです。それぞれの特徴を見てみましょう。

●ソニードライブ

それほど高性能ではありませんが，それでも非SCSI HDよりも高速だったりします。安定しており，ユーザーの満足度は高いようです。

●オリンパスドライブ

かなり高性能です。プログラムで制御すると，エラーチェックがいまいち甘めのようです（実害はないか？）。

●富士通ドライブ

小型軽量高性能です。基本的にはかなり速いのですが，ほっておくとパワーセーブモードになり電源が切れます。DIPSW2-4 OFFで解除できますが（MOを分解する必要があります），それでもヘッド退避しているのか，しばらくほっておくと再アクセスに3秒くらいかかります。

●松下ドライブ

基本性能は良好です。ディスクローディング時にちょっと時間がかかります。他機種より熱を多めに発生します。

＊　＊　＊

主なメーカーは以上のとおりです。ドライブ自体の性能というのが製品の性能の大半を占めますので注意が必要です。

接続

よくある「〜接続キット」というのは特に必要ありません。ただし製品によってはケーブルがついていないことがあったり，MOメディアが付属していないこともありますので，あらかじめ確認が必要です。

すでになんらかのSCSI機器を使っているという人（大半と思うが）は増設すると不安定になることもあります。

接続して素直に動けばそれで問題はありません。素直に動いてくれないとき，たとえば，単独でつなげば動くのに，起動時に

倍密ドライブについて

噂されていた倍密MOドライブが発売され始めました。現状では121Mバイトで不足することはあまりないのですが，今後，単密ドライブと倍密ドライブではほとんど値段が変わらないと思われるので，倍密型が急速に浸透していく可能性もあります。当面は必要なさそうでも，どのようなものなのかということくらいは確認しておいたほうがよいでしょう。

フロッピーディスクの倍密とは少々違います。実は磁性面での基本的な記録密度は引き上げられていません。

ここでとられたのは磁性体などは従来のもののままで，内周と外周の記録密度を一定にしているというものです。これまでのMOでは角密度が一定だったので内周でのセクタと外周でのセクタでは広さがまるで違っていました。外周部分ではかなりもったいない使い方をしていたことになります。これを同じ広さに揃えると結果的に高密度化がなされるわけです。

通常こういったことをやるにはモーターの回転を変えるなどの細工を行うわけですが（MacintoshのFDやレーザーディスクのCLVなど），MOでは等速回転のままこれを行っています。よってハードウェアはほとんど同じものですみます。

これを実現するには，かなり高度なサーボ技術が必要になってくるのですが，いったんできてしまえばドライブの値段自体はほとんど変わりません。これから発売されるMOドライブは倍密と単密で同じドライブを使ってファームウェアのROMだけを変えていくのだろうという噂もあるくらいです。

角速度一定のまま線密度一定にしている関係でちょっと面白いことが起きます。つまり外周のほうがデータ転送速度が速くなるはずなのです。

＊　＊　＊

もちろん，従来の121Mバイトディスクのデータの読み書きにはまったく問題ありません。

価格的にも変わらないため次期標準となることは間違いありませんが，世代交代はどの程度普及していくかにかかっています。

従来製品がすでに無視できないくらい普及していますので，しばらくのあいだはデータ互換のために現在の121Mバイトタイプが多用されていくと思われます。

MOディスククリーナ

です。フォーマットにはHuman68k ver.3.0のFORAMT.Xを使用するほうがよいでしょう。なお，他機種で物理フォーマットしたメディアではFORAMT.Xで再フォーマットしても容量が同じにならないことがあります。メディア交換の妨げにはならないのですが，同容量のHDとのあいだでDISKCOPYができなくなります。

FORMAT.Xを使った場合，装置初期化には10分間かかります。決して中断してはいけません。途中でエラーなどが起こると初期化が不完全になります。

領域確保の際は必ず121Mバイト1パーティションのみで確保します。特別な用途（とにかくドライブ数がたくさんほしいときとか）で使うとき以外は必ず1パーティションで使うようにしましょう。

本来ならばIBMフォーマットを使用するのがよいのですが，FORMAT.Xのサポート外ですし，特殊なドライバが必要でかなりアクセス速度が落ちてしまいます。

認識されていない。なにか作業をやっているとエラーが出る，アクセスするとデータが壊れるといったときには接続を確認する必要があります。

どうしても不安定なときは，SCSI機器の配列を変えたり，さまざまなことをやってみて動く組み合わせを探します。後ろにもう1台なにかつないだり，ターミネータをはずしてみたり，ターミネータの代わりにSCSIケーブルをぶら下げたり，ちょっと理屈にあわないようなことでもやってみるとなぜか動いたりします。機器の相性もあるようです。

また，SCSIケーブルのインピーダンスが異なるとうまく動かないことがあります。SCSIケーブルには抵抗値の高いものと低いものがさまざまあり，機器の前後でばらつきがあると動作しない機種もあります。怪しいケーブルは換えてみてください。

一般に，本体HD内蔵機種のほうが不安定なようです。

フォーマット

ドライブにはハードディスクモードと光磁気ディスクモードのいずれかに設定できるものがあります。X68030ではどちらのモードでも使用はできますが，必ず光磁気ディスクモードで使用するようにしてください。

ハードディスクモードでフォーマットされたメディアと光磁気ディスクモードでフォーマットされたメディアは違います。光磁気ディスクモードで使用していても，メディアをフォーマットしたときにハードディスクモードだった場合，そのメディアはハードディスクに認識されます。

これがあっていないとメディアの入れ換え時に不都合が起こります。

また，フォーマッタによって微妙に容量などが違う場合がありますので注意が必要

*　　*　　*

SCSI機器は起動場所に関わらずIDの小さいものが若いドライブ名になります。環境を持ち歩く場合，最低限，相対パスを使用して，ドライブ名に依存しない起動システム作りを心がけてください。

PATHや環境変数は，

●SET1
　SET BOOT=　　（改行しないこと）
●SET2
　SETENV
●AUTOEXEC.BAT
　COPY SET1 SET1.BAT
　CD >>SET1.BAT
　SET2 >>SET1.BAT
　SET1
●SETENV.BAT
　PATH=%BOOT%;%BOOT%BIN;
　　⋮

のようなファイル群を作成しておけばバッチ処理だけで設定できます。

クリーニング

エアフィルタを交換する必要がある機種はもちろん，メンテナンスフリーの機種でもある程度の手入れが必要になることがあります。

ひとつはレンズのクリーニングです。これはCDなどと同じ考え方ですね。

クリーナはMOメディアと同じ形をしており，ヘッドが移動する場所にブラシがついているだけという簡単な構造です。

もうひとつメディアのクリーニングというものがあります。これはMOメディアの表面についた汚れを取り去るものです。

光を使ってデータを読み出すのですからディスクの表面は汚れを嫌います。

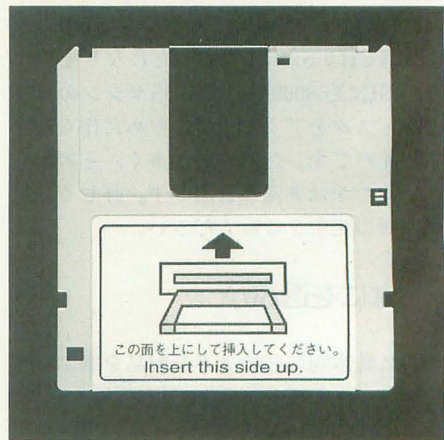
MOはフロッピーディスクと違い，表面に保護層があるためクリーニング液をつけたクロスで拭き取ることができます。使用環境によってはかなり汚れることがあるようですので気をつけましょう。

以前にも書かれていましたが，MOの性能にははっきりした法則があります。あなたが持っているMOの性能をもっとも安価に上げる方法は「友達にもMOを買わせる」ということです。この場合，友達とはX68000ユーザーでなくてもかまいません。

目の前でMOメディアのシャッターを開けてみせ，虹色の円盤を披露します。そして，おもむろに円盤の上面に指を乗せてペタペタ指紋をつけるという芸があります。実はMOの記録面は下側ですから，上面（ラベル側）にはいくら触っても支障はありません（汚れを嫌うメディアですから保障はしませんが）。

CGAマガジンに収録されている動画データなどをAMIファイルに落として連続再生するというのも効果的でしょう。

メディアはフロッピーのように持ち歩き，小型のものならドライブごと持ち歩いて颯爽と使います。

では，よいMOライフを。

MOヘッドクリーナ

松下ドライブを使ったMOユニット

汎用MOユニットを接続する

Kioi Makoto 紀尾井 誠

MOはちゃんとSCSIした製品が多いのがうれしいのですが，X68000の場合，MOの接続では素直にいかないことがあります。ここでは普通にはつながらない汎用MOを接続してみましょう。

とりあえずつないでみる

ここではジェフのSPINNER SP-120MOを紹介する。

あらかじめいっておくと，このドライブは特にX68000対応というわけではない。汎用のMOというか，わざわざ機種を特定していない代わりに一般的な手順で接続されることを要求するドライブである。世の中にはローカルなSCSIがたくさんあって，いちいち対応していられないということだ。

X68000のものはかなり素直なほうだが，MOのINQUIRYに関しては致命的な欠陥が露呈している。よって，まったく素直なMOドライブはX68000に接続できない状況にある。

SP-120MOもX68030で使うにはなにも問題ないが，X68000だとそのままでは認識してくれない。そこで，それに対応するフリーソフトウェアのドライバ，すなわちINQPATCH.XかSxSIのいずれかを使う必要がある。

ここではそれらを使って，このMOドライブをX68000に接続できるように設定してみることにしよう。

●INQPATCH.Xを使った場合

INQPATCH.Xは電脳倶楽部vol.52のものを使用した。TAKERUでも入手できるはずだ。このドライバはSCSIドライバがデバイスから受け取るINQUIRYコードを差し換えてしまうものだ。

IDが3だった場合ならCONFIG.SYSに，

```
DEVICE=INQPATCH.X -S 3 $1
DEVICE=SCSIDRV.SYS /ID3
```

という設定を加える。

Human68k ver.3.0ではSCSIDRV.SYSは必要ないし，そもそも付属していない。しかし，パッチを当てたあとにドライバを組み込まなければ意味がないので，Human68k ver.2.0のシステムディスクから引っ張ってくること。ちょっと危なっかしいが，これでも一応動いているのだからよしとしよう。

●SxSIを使った場合

SxSIは秀和システムズのフリーウェア本「X680x0コレクションACE」に収録されたものを使ってみた。

SxSIをSCSIボードなしでSCSIハードディスクをつなぐシステムとしか考えていない人もいるみたいだが，SCSI使用者にとっても有用なものである（なければないでかまわないが）。

この場合は，CONFIG.SYSに，

```
DEVICE=SCSIIOCS.SYS
DEVICE=SCSIHD.SYS ID=3 MX=1 X68SCSI.BRP IBMSTD.BRP
```

のように設定することになる。

SxSIでは容量の違うパーティションを許容し，しかも複数ドライブを設定することができるのだが，常識的には使わないほうがよい機能だといえるだろう。それなりに便利なのだが，わざわざローカルな仕様にすべきではない。

SxSIではIBMフォーマットのMOが作成できるというのはうれしい。逆に，Human68kのフォーマットが標準と少し違うので注意が必要なこともある。

＊　＊　＊

これらの設定で汎用のMOドライブが使用可能となる。この設定でやればちゃんと光磁気ディスクモードになるはずだ。

ドライブについて

最後にドライブ自体の紹介をしておこう。

SP-120MOでは1993年9月号で紹介されていたロジテックのドライブユニットと同じ松下製のドライブを採用している。ファームウェアのバージョンも変わっていないようだ。

動作を見ると3000回転にしては実に速い。操作感もよく，動作も安定している。

回転速度が最大の性能要因となる光磁気ディスクで，3000回転ながら3600回転のものとそれほど遜色のない速度を出すドライブなのだ。

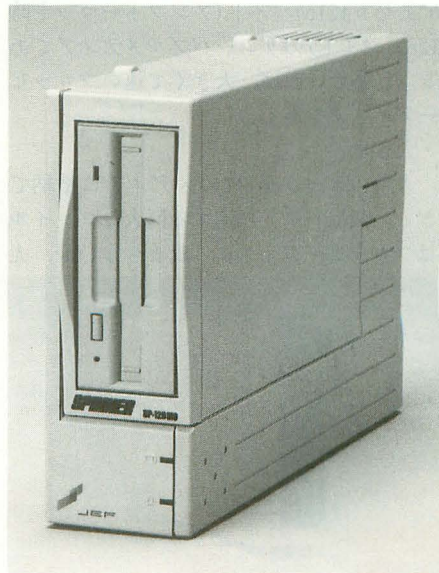
SCSI機器にとって安定していることというのは重要である。なかには一緒に接続する機器やケーブルの状態，接続時の配置順番でうまく接続できなくなる製品だってあるのだから。

個人的な意見だが，やはりかなり熱を持つのが唯一の難点といえるだろう。これくらいの熱でメディアが壊れるということはないとは思うが，発熱する電子機器というものに対してはどうも不信感を持ってしまうのだ（理由はない）。

「なんでこんなに熱くなるんでしょう？」

「なんせ殺人光線ですからねぇ」

いくらレーザーだからといって，読み込み時は低出力のはずなんだが。

壊れたという話は聞いたことがないので，これはこういうものだと納得するしかないのだろうか。

SP-120MO　148,000円（税別）

CS-M120 Filo(コパル)

3.5インチ光磁気ディスクドライブの活用

Tan Akihiko 丹 明彦

低価格で高性能かつ超小型，X68000専用モデルも現れたコパルのFilo。丹氏もユーザーのひとりです。さまざまなマシンを操る場合の使いこなし方を見てみましょう。

それは昨年の夏，SLASHシステムの開発で編集室に横内氏と缶詰め状態になっていたときだった。ご存じのようにSLASHはでっかい(いまはもっと大きいけれど)。2人が2台のマシンで作業をし，横内氏が新しいバージョンを作ると私がそれにあわせてCライブラリを書き直すという調子で開発を進めていたのだが，ソースコードやデータの受け渡しにフロッピーディスクを使っていたのではあまりにもかったるい。

が，世の中はよくしたもので，おりよく光磁気ディスクの特集が組まれていて，各社のドライブがマシン室にあったのだ。で，さっそくロードテストと称して使わせてもらい，おかげでSLASHの開発もより順調にいったというわけだ。このとき私が気にいって，のちに購入したのが本稿でご紹介するFilo(ファイロと読む)である。

光磁気ディスクの特徴

Filoは3.5インチ光磁気ディスクドライブで，いまさらいうまでもないが，1メディアあたり121Mバイト(アンフォーマット時128Mバイト)のリムーバブルメディアである。乱暴にいえば，大きくて速いフロッピーディスクである。

・大容量

大量のデータを1枚のメディアに格納できる。数Mバイトに達する巨大なファイルはフロッピーディスクに収まらないが，光磁気ディスクなら分割することなく格納することが可能である。

・安価

ドライブは10万円そこそこ，メディアは1枚あたり3500円程度で一部では3000円を切るものも出てきている。計算すればわかるが，容量あたりの単価はフロッピーディスクよりも安い。

・高速

遅いハードディスクという程度には速い。むろんフロッピーディスクよりはずっと速い。読み込みはハードディスク並み，書き込みはやや遅め。

・携帯性

メディアはちょっと厚手の3.5インチフロッピーディスクといった感じで，ポケットにも入るサイズである。

では，ユーザーの観点から見た光磁気ディスクを使うメリットを考えてみよう。

・でかいデータの運搬

大容量性と携帯性から，当然のように考えつくことである。巨大なプログラムやデータを，いちいち圧縮したり分割したりすることなく持ち運べるのは楽である。

・環境の運搬

光磁気ディスクはそこそこの高速性があるので，ハードディスクの代わりに使える。システムを入れておけば起動ドライブとしても使える。事実，私はX68030Compactを購入した時点で160Mバイトの内蔵ハードディスクが入手できなかったため，しばらくは光磁気ディスクだけで運用していたのだ。それを持っていけば，よそのマシンでも自分の環境で使える。また，万が一ハードディスクが壊れたときに備えてこういうディスクを用意しておくのもいいだろう。

・バックアップ

大容量だからハードディスクのバックアップとしても使える。バックアップに使うコマンドは「copyall /t」である。フロッピーディスクを使った場合のようにディスク交換の嵐になることもなく，ストリーマと違ってランダムアクセスもできる。

Filoの印象

このようにおいしい光磁気ディスクだが，なぜ私がFiloを選んだかというと，デザインが気にいったからである。とにかく筐体がコンパクトにまとまっているところがいい。SCSIコネクタはハーフピッチなので，ケーブルやターミネータを買ってくるときは注意が必要だ。

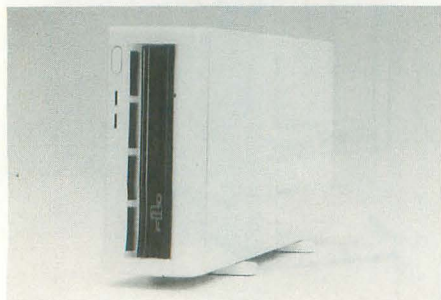
ただ，デザインという点では私はちょっと恨みがあって，私が白いFiloを購入した直後にX68000シリーズ用の黒いFiloが出たのだ。それはないだろうと思ったものだ。

もちろん，デザインだけで選んだわけではない。ドライブの特性が素直でどのコンピュータとでも接続できるというのは前提条件であったのだ。

メディアを入れる部分の蓋の形がちょっと変わっていて，引っ張って開けるタイプ。はっきりいって使いにくい。最初は苦労したが，いまでは片手でひょいひょいとメディアを挿入できるところまで極めた。あと，メディアが入っているかどうかのインジケータがないのは問題かもしれない。アクセスランプ(赤)を2色(赤/緑)にしておいて，メディア挿入時は緑，アクセス時に赤を点灯させるというデザインがよいのではないかな。LEDを増やせないなら，蓋を半透明にするだけでもかなり違うだろう。

電源スイッチが背面にあるが，特に困らない(ケーブルなどが邪魔にならないよう配置されている)。

ドライバになにを使うか?

特別なドライバは必要ない。が，私はSxSIを使っている。標準のSCSIドライバよりもいろいろな点で柔軟なので重宝する。

ひとつはIBMフォーマットの光磁気ディスクも利用できること，もうひとつはFiloを常に光磁気ディスクモードで(ハードディスクモードでなく）使うことができることである。

IBMフォーマットは他機種との共通フォーマットとして便利である(これについては後述する)。ファイル名が大文字のみで8文字＋拡張子3文字ということを除けば，Human68kのドライブと同様に扱える。

光磁気ディスクモードのうれしい点は，なんといっても電源OFF時にオートイジェクトしてくれることだろう。Filoは底面ロータリースイッチでハードディスクモード(ロータリースイッチが0を指す)/光磁気ディスクモード(同じく1）の設定ができる。自分のドライブをどちらに設定していたか忘れていたのでいまひっくり返して覗いてみたら0になっていた。つまりハードディスクモードなのだが，SxSIで登録すると光磁気ディスクドライブとして認識してくれている。DRIVEコマンドで見てもきちんと光磁気ディスクと表示される。

SxSIを使うときのコツは，「メディアを挿入しないで起動する」，これだ。SxSIをインストールして，光磁気ディスクドライブにメディアを挿入しないでシステムを起動すると，Human68kは光磁気ディスクをSCSIデバイスとして認識しない。がSxSIは認識する。そしてHuman68kフォーマットの光磁気ディスクを挿入すればHuman68kフォーマットのドライブとして，IBMフォーマットの光磁気ディスクを挿入すればIBMフォーマットのドライブとして機能する。リセットの必要もない。はっきりいってものすごく便利である。なおこのときはどちらのモードでもドライブ名は同じになる。

SxSIはIBMフォーマットの光磁気ディスクを利用するためにインストールしたのだが，いろいろとうれしいボーナス的な機能があるのである。

他機種との接続

私はX68000/030をメインに使っているが，AMIGAやMacintoshも使うことがある。CGなんて因果なものをやっていると，扱うデータは巨大になる。またマシンによって得手不得手もある。モデリングとレンダリングはAMIGAで，文字入れやレタッチはMacintoshで……といった分業を手際よくこなし，マルチプラットフォームを器用に渡り歩くのがいまどきのデキる計算機使いの必須用件といえよう。そんな用途に光磁気ディスクはうってつけだ。共通フォーマットはIBMフォーマット。

最近のAMIGAにはCrossDOSというMS-DOSファイルシステムのエミュレータが標準装備されており，IBMフォーマットのフロッピーディスク(2DD）を通常のAMIGA DOSのFDと同様に扱うことができる。で，このパラメータ指定を少し変えるとIBMフォーマットの光磁気ディスクも扱える。私はAMIGA1200にSCSIつきのアクセラレータを登載し，そこにFiloを接続している。AMIGA自身でフォーマットはできないが，他機種でフォーマットしたものを使用すれば大丈夫（ちなみにAMIGAで光磁気ディスクをフォーマットしようとしてハードディスクを飛ばした人が2人いる)。

Macintoshは，標準ではIBMフォーマットのメディアを扱うことができない(Apple File Exchangeはファイル変換ユーティリティにすぎない)。そこでIBMフォーマットの光磁気ディスクを利用するためのドライバを別途入手する必要がある。私はDOS Mounter Plusという製品を愛用している。IBMフォーマットの2DD(720Kバイト)，2HD(1.44Mバイト)，そして光磁気ディスク(121Mバイト)を，Macintoshの通常のボリュームとまったく同様の感覚で扱うことができるのだ。フリーソフトウェアにも同様な機能のものがある。

なお，FiloはMacintoshとも相性がよい。光磁気ディスクをMacintoshフォーマットで使用することもある。フォーマッタにはSilverliningを使用している。DOS Mounter Plusを組み込み，メディアを挿入しないで起動するとMacintoshフォーマットとIBMフォーマットを同一のドライブでリスタートすることなく扱えるのはX68000/030でSxSIを使った場合と同様である。

Filoを使用して

信頼性は問題ない。最初はおっかなびっくり使っていたのだが，きちんと使っていて読み書きのエラーを出したことはない。メディアをクラッシュさせたこともない。メディアがクラッシュするのはたいていメディア交換時であるから，次の習慣は大切である。

・メディアを抜く前に「cd ¥」する
・BREAKキーでバッファをフラッシュする
・新しいメディアを挿入してからは安定するまでアクセスしない

耐久性についても問題ないとみていいだろう。半年ほどほぼ毎日使っているが元気に動いている。なおFiloはメンテナンスフリーで，エアフィルタの交換は必要ない。

速度についてはおおむね不満なし。「おおむね」というのはときどき反応が遅くなるから。Filoのドライブは，「地球にやさしい」なんていう近頃の風潮を反映してか，しばらくアクセスがないとメディアの回転が停止する。この状態からアクセスにいくとけっこう(約5秒）待たされる。この機能は内部のディップスイッチを設定することで殺せるが，私はX68030Compact内蔵の160Mバイトハードディスクをメインに使っていて光磁気ディスクは運搬用と割り切っているので，中はいじっていない。

ただ，それでも富士通のドライブの場合は定期的に機械的な微調整を行うらしく，連続してアクセスしていてもときどき遅くなることがある。

＊　＊　＊

蛇足だが，冒頭のような複数人による開発現場では，X68000/030だから光磁気ディスクが最善の策だったのだが，MacintoshやUNIX系のワークステーションならネットワークを使ってファイル共有するのだろう。

が，大学の新学期も始まり，横内氏はちょいと遠くにいってしまったので，光磁気ディスクによるファイルのやりとりが役に立ちそうである。それにはまず彼に光磁気ディスクドライブを買ってもらわないとね。

私の環境

本体：X68030 Compact，無改造。
数値演算コプロセッサ：MC68882。
メモリ：8Mバイト(標準4Mバイト＋専用増設メモリ4Mバイト)。特に不足を感じたことはない。
ハードディスク：内蔵160Mバイト。さすがに高速(ギガバイト級のドライブにはかなわないが)。
モニタ：CZ-613D。X68000 PRO II用を流用。またはテラドライブのモニタ。買って1年になるが最近24kHzモードの動作が不安定になってきた。
MOドライブ：Filo。本文を参照してほしい。おおむね満足して使っている。
プリンタ：キヤノンのBJ-10vを使っていたがシートフィーダが使いものにならないので同社のBJ-220JCに買い換えた。
モデム：Panasonic TO-703B。一応FAXモデムだがFAXソフトがPC-9801用なので単なるモデムである。14400bpsだがその速度をフルに使ったことはない。
MIDI：MIDIボードはシステムサコムのSX-68M II。楽器はローランドのCM-500。最近火が入ってないなあ。

[特集] X68000と仲間たち

転ばぬ先の杖

満開式磁帯駆動装置壱號

Tomonaga Takeaki 友永 健明

ともすれば怠りがちになるハードディスクのバックアップ。HDが1Gバイトを超えると切実に必要となるのがストリーマ，すなわち大容量バックアップ専用のメディアなのです。

バックアップ取ってますか？

いまさらいうまでもなく，すでにかなり多数の人がハードディスクを使っていることと思います。ハードディスクがあって初めてわかる快適なパソコン環境というものを満喫していることでしょう。快適であればあるだけ，ハードディスクへの依存度が高くなります。そうなるとハードディスクユーザーがもっとも恐れるのはハードディスクのクラッシュではないでしょうか。

それほど高い確率ではありませんが，それでも毎年確実に何件かはハードディスククラッシュの話を耳にします。もともと振動や埃を嫌う超精密機器なのですから（しかも可動部がある），使っているうちに一定の確率でクラッシュが発生してもそれほど不思議なことではありません。

しかし，100Mバイト以上のハードディスクを使っていて定期的にバックアップを取っている人はどれくらいいるでしょうか？ドライブが大容量になり，使用量が大きくなるにつれ，ちゃんとバックアップを取る人は少なくなっているように思われます。

普段使っているシステム環境が突然なくなってしまうと，非常に大きなダメージを受けます。これは物質的なものよりも，むしろ精神的なもののほうが大きいかもしれません。

ハードディスクだけではありません。

MOもかなり普及してきていますし，FD代わりに常用する人も増えてきました。となると今度はMOのバックアップという問題も現れてきます。5月号の「言わせてくれなくちゃだワ」の列島縦断はみ出しでMOが飛んだという文章が掲載されていましたね。

システムが飛ぶことも悲惨ですが，せっかくMOにまとめたデータがまとめて消えてしまったらこれは悲劇以外のなにものでもないでしょう。

そこでストリーマ

なんやかんやで信頼性の高いメディアが望まれるところですが，長期保存などを考えるとどれも確実とはいえません。

ではどうすればいいかというと，もっとも確実にストレージの信頼性を上げる方法は「何度もバックアップする」というものです。それには高速でバイト単価の安いメディアが望ましいのはいうまでもないでしょう。

となると注目されるのはストリーマです。

ただ，誰でもストリーマでバックアップを取るのが最良かというと，そうとも限りません。ハードディスクの1パーティションが100Mバイト以下ならMOを使ってバックアップを取るほうが使い勝手がよいでしょう。ランダムアクセスできる分，MOのほうがファイル整理が簡単です（人によっては整理してからバックアップするみたいですが）。

MK-DS1　125,000円（税別）

しかし，ドライブが大容量化してくるとMOでも持て余すようになります。

1Mバイトあたりの単価を見てみましょう。

FD	50円
3.5インチMO	24円
DATストリーマ	0.1～0.5円

といった具合です。ストリーマの場合，自動的に圧縮されますのでバイトあたりのコストは変動します。オーディオ用DATテープが使用できるのでメディアの入手法にも問題ありません（データ用カートリッジは秋葉原でも探すのが困難）。

満開式磁帯駆動装置壱號

ということで今回紹介する満開式磁帯駆動装置壱號（MK-DS1）ですが，この製品に使用されているドライブユニットはDATテープを使用できるものです。

ストリーマには，ほかに専用のテープを使ったものや8mmビデオのテープを使ったものなどがあります。DATストリーマの記録方式はもともと3重のECC（エラー訂正符号化）を行っているDATフォーマットの，さらに信頼性を上げたものと考えてよいでしょう。テープも非常に小さく保管場所を取りません。

テープドライブユニットはヒューレットパッカードの製品を使用しています。この製品の特徴はデータ圧縮機構を備えていることです。これにより，容量，転送速度などを通常のものに比べて2～4倍に高めています。バックアップ速度が速いというのはうれしいものです。

転送が速くなると，つながる先の機器にもそれなりの速度が求められるのですが

(ランダムアクセスできないため)，この製品では内部バッファがそこそこ大きいので10MHz機でもまず問題はないようです。

気になるメディアの容量は60mテープで最悪1.3Gバイト，最高で5.3Gバイトになります。いわゆる薄手テープ(90m)にも対応しており，それを使った場合は最悪1.9Gバイト，最高で8Gバイトの容量になります。圧縮が絡んでくるため，データの内容によって圧縮率は変わり，一定ではありません。テープ自体は等速で動いていますので，当然，圧縮率が高いほどデータの転送速度も高くなります。

今回使用したのはスタンドアロンのドライブですが，TOWER JACKユニットに増設する満開式内蔵型磁帯駆動装置壱號も用意されています（税別138,000円)。

テープというもの

ストリーマを使ううえでは，媒体がテープであるということを忘れてはいけません。たとえば，常識で考えても途中で変なデータを上書きするととんでもないことになるのはおわかりでしょう。常に「これはテープなんだ」というイメージを持っておくことが重要です。

いくつか注意点を挙げてみましょう。

1) 基本的に上書きはできない

書き終わったところでそのテープは終了したことになります。基本的に1メディアに1ファイルなのです。書き足すときはいちばん後ろに「追加」という手順になります。

2) 内部はブロック化されている

1ファイルということではイメージがわかないと思います。内部はブロック化されており，その構造を把握しておくことが必要でしょう。

テープには必ず先頭と終端があります。先頭はたいていテープを巻き戻した位置から始まっていますが，終端はテープの最終位置とは限りません。データを書き足すごとに後ろに下がっていきます。新しくデータの書き込み（追加）を行うと，データのブロックを書き込み，最後にインデックス用のブロックを書き込みます。

3) インデックスのブロックにも大きさがある

操作が終わったときインデックスの先頭にいるか終端にいるかに注意が必要です。

3回バックアップを取ったテープがあるとして，2回目のデータを取り出したいとしましょう。読み書きは現在のヘッド位置から作業を開始しますから，あらかじめヘッドを先頭から次のインデックスまで移動しなければなりません。

ヘッドを移動するためのコマンドがMTです。

MT −1

でひとつ前のインデックス終端，

MT ＋2

で2つ後ろのインデックス終端へヘッドを移動します。

DATのサーチと同じですから，そこそこ高速です。

＊　＊　＊

DATストリーマ用カートリッジ

音声と同じ媒体に記憶するという意味では，X1などのカセットテープを思い出したほうがよいかもしれません。

パーソナルユースで使用されたカセットテープとUNIXなどのバックアップメディアとして使われたテープではやはり使い勝手が違います（カセットテープのほうが上)。現状でははっきりしませんが，サポートはNIFTY-Serveなどでも行われますので今後メンテナンス用のツールが追加されるかもしれません。

テープというとどうも遅いものというイメージを抱いてしまいますが，さすがにSCSI機器というべきか，シーケンシャルな転送速度はかなり高速です。実際，驚異的なことに3.5インチMOよりも転送速度は上です。

使い方

このストリーマを扱うためにはテープ用のデバイスドライバを組み込む必要があります。

ただし，Human68kのファイルシステムではランダムアクセスメディアしか考慮されていませんから，このドライバを使ってもドライブ名でディレクトリを取ったりということはできません。

ファイルシステムに関する考え方自体がまったく違うメディアですから，テープにアクセスするには，さらになんらかのツールが必要になります。この製品ではGNUのtar（Tape ARchiver）を使っています。

ドライバなどのソフトは汎用のもので，このストリーマ以外にもEXABYTEタイプのストリーマなどでも使用できるものです。ほぼフリーソフト扱いになるので，違う形式のドライブを入手できる人は試してみてもよいでしょう。

使い方は2とおり考えられます。ひとつは，もっぱらtarを使ってハードディスクのバックアップ専用にするというものです。普通はこれだけでもストリーマ導入の理由になります。

もうひとつは大容量データ媒体として活用する道です。多少SCSIとプログラムの知識が必要になりますが，ランダムアクセスはできなくても，Gバイト単位のリムーバブルメディアというのは可能性を秘めたメディアだといえるでしょう。

こういった用途として満開製作所で用意していたサンプルテープがありました。それは512Kバイトのグラフィックデータを G-RAMにベタ読みするというものでしたが，表示は1枚当たり1秒弱，だいたい700Kバイト/秒くらいの転送速度に相当します。SCSI対応のアニメーションツールAMIで想定されている転送速度は512Kバイト/秒ですから，ちゃんと対応すれば1時間のアニメーション連続再生も不可能ではありません（読み書き部分で多少修正が必要と思われます)。ビデオテープへのコマ録り機材と比較してはいけませんが，それに近いことはできるかもしれません。

ちなみに，使っている機材はオーディオ用DATと同様のものですが，非常に残念ながら，音声は入出力できません（サンプラー代わりになれば面白かったのですが)。

＊　＊　＊

すでに世の中にはグラフィックデータなどをストリーミングテープでやりとりするという集団もあるようです。

そろそろGバイト単位のハードディスクを考えている人もいるかもしれません。すると，その次のターゲットはきっとストリーマということになるのでしょう。

NSカルコンプ社Drawing Slate

さらにに進化したサイバー絵筆

Kawahara Youi 川原 由唯

グラフィッカー御用達の手描き感覚入力デバイス，それがタブレットです。スタイラスペンの接触でアモルファス材の表面に起こる電磁誘導をもとに座標を割り出す……実はかなりハイテク製品だったりします。

CGといえば，なにかにつけて「冷たい」だとか，「固い」だとか，無機質な印象しかもたれていませんでしたよね。多少なりとも最近のイラストCGを知っている方ならそんな印象はもたないと思うのですが。でもやっぱり，なにも考えずにCGを描くと，ベタ塗りアニメ調の絵やつやつやテカテカなプラスティックイメージになりがちなんですよね。

なにがいけないって，基本的にオン/オフしかできないデジタルな入力装置が悪いんじゃないでしょうか。

さてさてタブレットは，アマチュアのCG絵描きのあいだでも一般的な入力装置になりました。手に届く価格帯に落ちてきたからなのですが，X68000の場合はMATIERの普及も一端を担っていると思います。マウスコンプレックスを持っているCG絵描きにとっては，筆の印象を容易に表現できるソフトの登場は救世主に思えたものでした。そして手描きの再現をするならやはりどうしてもタブレットが必要です。

そんな感じでタブレットとMATIERは発売当時飛ぶように売れたと聞きます（嘘じゃないぞ）。

新画板，ドローイングスレート

今回紹介するのは，Drawing Padでお馴染みNSカルコンプ社のデジタイザ「Drawing Slate」です。X68000とはRS-232Cで接続します。

このDrawing Slateは軽量小型でほとんど場所をとりません。本体サイズが約280mm×249mm×6mm，描画有効範囲が約230mm×153mm（実測値）です。

信じられないことにコントローラ部さえもこの6mmの薄っぺらい本体に内蔵されています。12Vの（ちょっと大きめな）電源アダプタは必要ですが，接続コネクタはRS-232Cケーブル部なので，手元が煩わしくなることもないでしょう。使わないときはマウスパッド代わりにしてもいいかもしれません。

ケーブルも，強度に不安を感じてしまうほど細く軟らかいものなので，取り回しも効きます。よくもまぁ，こんなにコンパクトに設計できたものだと感心してしまいます。

また，旧Drawing Padは縦に長く，我らがX68000で使うには少々無駄がありました。このDrawing Slateは物理的にも横に長い描画エリアなので，場所をとることなく有効に使えます。

表面は梨地の施されたビニールシートがかかっています。これをめくって，紙に描いた下描きをはさんでおけばトレースもできます。裏面はゴムコーティングされており，滑り止めとして有効です。

Drawing Slate　74,800円（税別）

専用のペンは，もちろんコードレスです。人差し指付近に2つのスイッチ（ボタン）がついており，それぞれに別の機能を割り当てられます。なお，ペンにはボタン電池を使用しています。

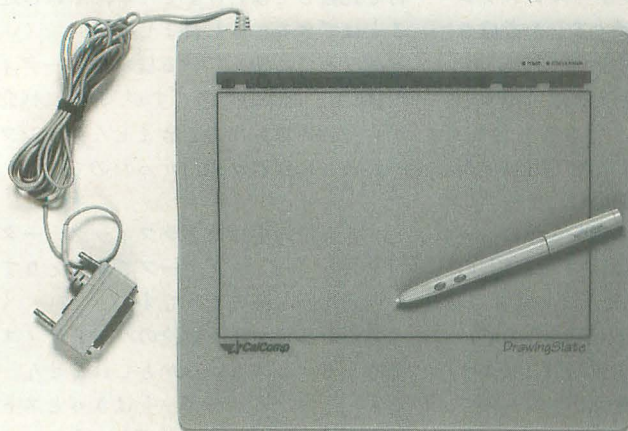
そして驚きの価格，定価でなんと8万円を切っています。コストパフォーマンスを考えれば，これは絶対お買い得。

カルコンプのタブレットは，もともと“筆圧”や“筆の傾き”を感知できましたので，ソフトウェアさえ対応できれば以前紹介したDrawing Padでも面白い表現ができました。

いまのところ，筆圧に対応している市販お絵描きソフトは，MATIERとHyperPixelWorksです。HyperPixelWorksのほうは，時間と資料の都合上，残念ながら試用することができなかったのですが，なんとこちらは“筆の傾き”にも対応しているのだそうです。これらを使いこなせるようになったときには，いったいどんな作品が生まれてくるんでしょうね？　少々怖い気もします。

というわけで，今回はMATIERの最新版を使って，ほんのちょっといたずら描きをしてみました。

MATIERについて

MATIER登録ユーザーには，いつものように最新版フロッピーが郵送で届いていることだろうと思います。この最新バージョンがバグフィックスとともに，Drawing Slateに対応したものです。Drawing Slate対応機能は，「追加機能（シェアウエア）」となっていますので，継続して使おうとする場合は，サンワードに代金（3000円）を振り込んでください。

試用してみましょう。

10MHzのX68000でも，筆の太さを極端に太くするなどしなければほとんど不満のないレスポンスでした。

筆圧感知モードで描画すると，温かさを感じさせる，鉛筆や水彩の「ほわ～」とした筆運びやコンテでスケッチをするような感覚で描けます。ブラシサイズを若干大きめに設定し，線幅モードを選択すると，ランダムな手描き感が大きくなって効果的です。

専用のペンには，人差し指のあたりに2つの小さなスイッチがあります。上にあるのが第1ボタン，下にあるのが第2ボタンです。

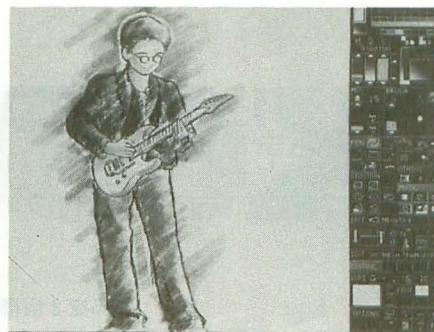
第2ボタンはマウスの右ボタンに相当するもので，描画中からメニューに移りたいときや，モードチェンジなどに用います。そして第1ボタン，これが今回の目玉で，これをオンすることで筆圧感知モードに入れます。

筆圧感知モードに入り，ペンでそっとタブレットの上をなぞると，薄く描画されます。徐々に力を込めていくと，それに応じて描画濃度が濃くなります。濃度はスプラッターブラシに対応しています。初めて触ったときはカルチャーショックでした(笑)。同じ濃さで描きたければ，もう一度第1ボタンを押します。

こんなことができる

MATIERでスケッチする

スケッチ画に最適

ちょっと太めの丸ペンや，ざらざらのブラシペンを使って描画すると，まるで本物の画材を使っているような線が引けます。黒で描画していると，本当に筆ペンで描いているような錯覚さえ感じてしまいます。以前冗談でいっていた「書き初めCG」もにわかに現実味を帯びてきたりして……。「鉛筆画“風”の絵を描く」だとか「水彩“調”の絵を描く」なんていう消極的な発想ではなくて，ちょっとペンさばきに慣れたら本当にこれ自体がスケッチとして十分活用できそうです。ますますX68000ノートの登場を望む声も増えそうですね。

オートペイントで使ってみた

筆圧感知や傾き感知はタブレットの特権。マウスではどう頑張ってもできないぞ。筆にできてコンピュータでシミュレートできないパラメータって，あとほかになにが残ってましたっけ？

愛のあるユニークで豊かなお部屋mkII

1993年8月号のお部屋紹介からそれほど変わってはいないのですが……。とりあえず据置型のHi-8ビデオデッキを買ったのと，SC-55mkIIが増えました。

●X68000 ACE-HD グレー
特に変わってません。内蔵HDナシ。メインメモリ6Mバイト。数値演算プロセッサ，SCSIボード増設。

●ITEC ITX640
なんと，悲しいことにSASIのHDが半身不随になってしまいました。前面の電源スイッチが壊れてしまったのです。でもHD自体の機能は大丈夫なので，コンセントの抜き差しをスイッチ替わりにして使ってあげてます。そろそろギガディスク購入も考えているのですが……。

●SC-55mkII (Roland)
某メーリングリスト関係の合宿で洗脳されて，思わず購入。いまのところ聴くの専門。そのうちにはデータ作りにも手を出したい。

●PC200 (Roland)
弾けもしないのにMIDI-キーボードもあったりする。MIDI関連は，基本的に98noteに接続してます。制御ソフトはフリーソフトのmimpiとか，市販ソフトのレコンポーザV2.5とか……。

●カラースキャナ GT-6500 (EPSON)
ついにSCSI接続しました。やっぱりSCSIは速いですよぉ。RS-232C接続時代は躊躇してしまっていた，フルカラー取り込みも苦になりませんから。

●タブレット SD-510C (WACOM)
筆圧対応が当たり前（？）の時代になってしまって，いまやこの機種も旧式なんでしょうね。別に不満はないですけど。

●モデム MD96FB5V (OMRON)
9600bpsも一般化しちゃいましたねー。買った当初は速い部類に入っていたのに。商用ネットも9600にちゃんと追いつくレスポンスが出るようになってほしいですよね。

●3.5インチMO RMO-S350 (SONY)
数週間前，ファイル転送中にSCSIケーブルを誤って引っこ抜いてしまい，100Mバイト近いデータを描きかけのCGもろとも昇天させてしまいました。ここまで激しくダメージをうけたのは初めてです。

●3.5インチFDD TS-3XR (TSUKUMO)
会社のUNIXや98関係とファイルをやりとりするときに使ってます。どうあがいても，これからは3.5インチの時代でしょう。

●カラーイメージユニット CZ-6VT1
あまり使ってません。なにか面白い使い方があったら教えてください。

●熱転写カラープリンタ CZ-8PC2(SHARP)
やっぱり最近もほとんど使っていない。

●PC9801 SX/E
主に通信とMIDI関係の制御に使用。メインメモリ4Mバイト増設。250MバイトHD内蔵。Windows? なんですかそれ。

おまけ

MATIERの追加機能に「オートペイント」ができました。これは写真や絵などを加工して油絵のような雰囲気にしてしまう特殊効果です(その昔，Oh!Xでも丹氏がプログラミングしていた，あれの高機能版です)。絵が描けなくても，写真とスキャナがあれば，それなりのアートができてしまいます。これもシェアウェアなので，継続して使う場合は代金を支払う必要があります。サービス精神旺盛なサンワードさんの良心に応えて，ぜひ送金してくださいね。

使い方。まず裏画面に加工したい画像を置きます。その後，スプラッターブラシなどで画面をなぞると，裏画面の色が筆のパターンで表画面に現れてきます。画面全体を自動描画してもかまいませんし，手で根気よく描画を繰り返してもかまいません。やがて裏画面が絵画のようなタッチに加工されて表画面に現れてきます。知ってる人は，ドラマ「あすなろ白書」のオープニングとかを思い浮かべてもらうとわかるかな。あとシャッセのCFとか（あれは本当の油絵だったらしいけど）。

ペンの調整によっては，色鉛筆や水彩のような雰囲気も出せる，なかなか面白い機能です。顔や目など，微妙な表情を壊したくないところは細かいペンを使うとよいでしょう。

加工する原画には注意してくださいね。自分で楽しむだけならかまいませんが，外に発表したりする場合は，原画の著作権，肖像権に触れないようにしなければなりません(今回のサンプルはもちろん私個人の作品だよ)。

ローテク工作実験室番外編

謎のPPIボードただいま見参

Taki Yasushi 瀧 康史

8255を2個搭載し，メカトロ制御からROMライタ，高速通信そしてPCM音源まで，ありとあらゆる用途に応えるPPIボード。セミキットタイプで自作派のための周辺機器です。

なんでせう？

Oh!Xの巻末に去年の年末から毎月広告が掲載されている科学工芸研究所のPPIボード。同じく掲載されているスキャナボードのほうは，なにに使うかはすぐにわかるものの，これっていったいなに？　って方が多いのではないでしょうか。

オプションの少ないX680x0で，なぜかこの時期になってX680x0に入ってきてくれたこのメーカー。せっかくだから大事にしてあげたいけど，いったいなにに使うのかなぁ……なんて人も多いはず。

「わけのわからないハードはとりあえず瀧か石上に渡してみよう」という謎の風潮から，なんでか知らないけど，やっぱり私に回ってきました。

まあ，面白そうなものだからよいですけどね。

最近どうもお葉書で，「初心者にもわかる記事を！」という要望が多いので，初心者にもある程度わかるように簡単に説明しておきましょう。内容が内容なので，あまり砕いて書けないのはご容赦ください。

パラレル入出力

まず，このボードはセミキットボードです。つまり，自分でハンダゴテを暖め，対応するソフトウェアを作る人にしか，存分に使うことはできません。ただ，どんなハードウェアでもそうですが，誰かがソフトウェアをちょっと作ってくれるだけで，「ハンダゴテの使い方さえわかれば」パソコン初心者でも使用することができます。たとえば，私がこれを使うソフトを作り，どういうハードを作るか指示し，同じようにハードウェアを作ることができるならば，初心者でも使用できるということです。

では，このPPIとはなんなのでしょう？

ひと言で説明を済ませてしまえば，これはパラレル入出力ポートです。「パラレルってなに？」という人がいると思うので簡単に説明をすることにします。

まず，コンピュータのデータの転送には，大きく分けて2つ方法があります。ひとつはパラレル転送，もうひとつはシリアル転送です。ポートというのは，港，つまり，コンピュータのデータの送り口ですね。

わかりやすく例を挙げましょうか。シリアルポートでもっとも有名なものは，RS-232Cです。パソコンが本当にローカルであったときを除き，たいていのコンピュータにはついていたものですね。また，楽器などを制御するMIDIも実はシリアルポートです。

一方，パラレルポートの例はプリンタポート，ジョイスティックポートなどです。X680x0の特徴ともいえるイメージ端子もパラレルポートですね。

さて，シリアルとパラレルの違いを挙げましょうか。ここで仮に16ビットのデータがあるとします。この16ビットのデータを別のコンピュータに転送するとしましょう。

シリアルの場合，16ビットを16個に分解し，1つひとつ，送っていきます。つまり，すべて送るには，16回もかかります。仮にひとつ送るのに，1msかかるとするならば，16msもかかってしまうのです。

一方，パラレルの場合，16ビットパラレルポートならば，1回ですべて送れてしまいます。理由はコード（信号線）が16本あるからです。先ほどと同様，ひとつのラインを1msで送るならば，1msで16ビットが送れることになります。

つまり，同じ転送速度ならば，シリアルよりもパラレルのほうが，ビットの分だけ高速に転送することが可能になります。

パラレル転送は速いのですが，その分，信号線を多く必要とします。今回のPPIボードは48ビットI/Oがありますので，グランドなどをあわせ，60本もの，信号線を利用しています。すなわち，その分お値段は高くなりコネクタ幅（面積）も広くなります。

しかしながら，パラレル/シリアルというのもいささか古い言葉なので，いったい，パラレルなのでしょうかシリアルなのかわからないものもいくつかあります。たとえば，16ビットのデータを4本の信号線に乗せて4回転送したときは，パラレルなのかシリアルなのでしょうか？　答えはパラレルなのでしょうが，こうなるとシリアルは1ビットパラレルということになりますよね。

PC-98x1などで割と有名なMAXLINKというソフトがありまして，最近Oh!Xでも由井氏が行っている仮想ドライブ実験に近いものだと思われます。2つのマシンをRS-232Cでつなぎ，それぞれのドライブを互いに増設ドライブのような感じで使ってしまうデバイスドライバなのですが，RS-232Cでつないでいるクセに異様に速いというのがこの製品の特徴のようです。

要はRS-232Cの使用されないエラービットを利用して4ビット（だったかな？）パラレルで転送しているんですよね……それぞれを38400bpsぐらいで転送できるならば，38400bps×4＝153.6kbps＝19.2kbpsものスピードで転送することができます(PC-9801ではマシンにより9600bps以上が裏技を使わないとうまくいかないので多分MAX LINKは9600×4でしょう。憶測なので信用しないように)。

道がはずれましたが，PPIボードは要は48ビットパラレルポートという代物です。いったいこれをなんに使うのかといわれれば謎です。

48ビットもバスラインがある……。なにに利用できるかは，まったくユーザー任せなボードといえるでしょう。なにしろセミキットなんですから。

製品説明

そうです。とにかくPPIボードというのはセミキットなのですから。そういう視点でいかなるものか，まずはマニュアル，そして基板の回路図をのぞいてみました。

さすがにここからは初心者向けには記事は書けないので，ハードウェア工作をある程度している人たちを対象とします。

マニュアルはそれほど厚くはありませんが，別用紙で回路図が入っています。GAL（ユーザー定義可能な論理回路素子）に関しては，論理がマニュアルに付録として記載されています。つまり，ボードを買った人は，このボードのすべての情報を得ることができるというわけです。これはとても素晴らしいことです。

I/Oポートの管理には，知る人ぞ知るi8255という，昔からのICが利用されています。実際に基板についているものはこれと互換性のあるNEC製のμPD71055C-10というものですが，高速でCMOS版というだけで同じものだと見てよいでしょう。

かなり昔のICですが，（昔は複雑だといわれてはいましたが）プログラマブルに操作ができて，なかなか多機能なので私もときどき使うことがあります。ちなみに，X68000でもジョイスティックの管理などに使っています。

このμPD71055C-10が2つ。

i8255の説明はこのあとの項に任せて次に進むことにします。

このボードにはランド（部品の取り付け場所）がいくつか用意されており，それらは自分で部品を購入して，ハンダづけをします。バッファなどもついていないので，自分で装着する必要がありますが，その分，自分でものを選べます。

コネクタは圧着コネクタが接続できる60ピンのランドが用意されていて，I/Oには入力用にLS245，出力用にLS541，またはトランジスタアレイとしてTD62083などを直接ハンダづけできます。また，直接I/Oに出力することもできますし抵抗によるプルアップ/ダウンのためのランドも用意されています。

これらはすべてマニュアルに記載されていて，必要ではないかと思われるすべての情報が解説されています。少し，蘊蓄めいたボード上の部品の説明など，デジタル回路を勉強する人にとって有用な情報もあるので，デジタル回路の勉強をしていてX680x0になにかオプションボードをつけてみたいという方には非常に有用なボードではないでしょうか？

私のローテク実験室でも，同様の機能のボードを作るつもりだったのですが（汎用入出力ポートとして），GALの配布，部品の多用化から，同じようなボードを作るくらいなら，このボードをそのまま利用するか，改造して利用するのが楽そうな気もしますね。

外販しているようなので，流通にお金がかかっているせいか，定価は22000円と，自作した場合に比べるとちょっと高めかもしれません（GALが焼ける人には，だが）。まぁ，一般人には不必要なボードともいえるので，しかたがないことなのかもしれませんが。

PPIボードに部品を装着したところ

i8255(μPD71055)とは

ここで少し，i8255というICについて説明することにしましょう。

コンピュータをなにかに接続するとき，先ほど説明したとおり，なんらかのバスラインが必要です。それはシリアルであったり，パラレルであったり，場合に応じてさまざまですが，たいていは性能，コスト，大きさなどを天秤にかけ，最良のものを使用します。

パラレルインタフェイスを使用するときはたいてい速度を重要視しなくてはならない場合です。その入出力が簡単であり，応用がきかなくてよいのならば，CPUのバスラインから直接抜き出し，なんらかのロジックを加え，引き回すことにします。場合によってはGALを利用して，ロジックを焼いたりして簡略化します。

しかしながら，インタフェイスがちょっと複雑になるだけで，使用するICが格段に増え，配線も複雑になり，ICが増える分消費電力も増えていきます。GALを多用することで，これらの欠点をある程度抑えることはできますが，GALソースをここに掲載しても，読者のいったい何人の人が同じGALを焼くことができるでしょう？

そこでi8255のような汎用的なパラレルI/Oを利用するというわけです。

同じような目的で作られた汎用パラレルポートは，私が知っている有名なものだけで3つほどあります。

ひとつはモトローラのMC6821。パラレルポートは8ビットが2チャンネルあり，MC6800，MC68000には簡単に直結できますが，MC68030に接続するには，ちょっと

PCMボードはちゃんとつくるぞ!

今回作ったD/Aコンバータだが，遊びで作ったものの，なかなか面白かった。本体内蔵のADPCM音源にはそろそろ限界を感じているからね。今回のPCMボードは音程も変えられないし，再生しているあいだはシステムがものすごく重くなっちゃうとか，ほかに同時になにもできないものだったけど，それでもリアル8ビットPCMってのは面白かったな。

しかしやっぱり8ビットPCMってのは，まだちょっとつらいところかもしれない。なんといっても低音域が腐るのは痛い。周りの人間も痛切にいっているので，最近，PCMボードを真面目に作り始めた。

スペックは最大48kHz，16ビットサンプリングなら問題ないだろう。データ量が膨大になって，保存場所はどうするのだ？　という話があるかもしれないが，それはMOで解決させてもらうことにする。DMA2を使うとゲームと相性が悪くなるからコントローラから落とさないといけないな。

まぁそのうち，モノになりそうなら，大々的に発表するからさ。大がかりなものなので1年近く製作にかかるかもしれないけど，そのときになったらよろしく。

音源側のボード

した努力が必要になります。つまりcompactXVIまでは簡単に接続可能ですが，X68030にはほかのCPU（たとえば80486）に接続するのと同じぐらいの手間がかかってしまいます。ちなみにこれはPIAと呼ばれています。

もうひとつはザイログのZ80PIOというもので，Z80には接続しやすくできています。たぶん，X1あたりには使用されていたのではないのでしょうか？　Z80用ということで，あまり使用する気が起きなくて調べていません。名のとおりPIOと呼ばれています。

そして，インテルの8255です。特に80系に使用されていますが，別にWRとRD信号が独立している程度で，特別に80系でなくては使用し難いというわけではありません。PPIといわれているものです。

3つのうちでいちばん利用されているものが8255ですが，その理由はこのICが安く，周辺装置側（ペリフェラル側）の入出力のビット数が多いという理由からでしょう。

なにぶん古いICで，私自身秋葉原の本屋を探したものの，8255そのもののマニュアルはありませんでした。おそらく，NECのV30あたりの周辺チップマニュアル集には，μPD71055として，詳細が掲載されているのでしょうが，こういうのは普通，絶対使わないようなほかのICのマニュアルと一緒に掲載されており，その本自体猛烈に分厚く，高いと思われるのでソフトウェアから必要とされるような情報はここで説明しきってしまいましょう。

このボードではCPU側の接続はすべてGALが行っています。ですから，このボードを利用したい場合，CPU側の接続は（興味があったらマニュアルのGAL論理を見て学習するのもよいでしょう）特に考えなくても構いません。

ペリフェラル側は，PA0～PA7，PB0～PB7，PC0～PC7と，3ポートに分かれています。この3つの8ビットポートをどう使うかは，8255のコントロールレジスタへの書き込みによって変えます。

8255のモードは3種類あり，それぞれ，モード0，モード1，モード2です。しかし，モード0以外の使い方は，PCが上位と下位に分離され，上位はPAに，下位はPBに従属されてしまいます。

また，モード0は単なるI/Oとして利用できますが，モード1は8ビット単位のハンドシェイクモードに，モード2は双方向ハンドシェイクモードになります。

ペリフェラルピン幅を生かし，I/Oとして使うならばモード0が妥当といえるでしょう。図2にモード0時のポートの書き出しについてまとめておきます。

このように最初に8255のコントロールレジスタに設定しておけば，ユーザーはそのままあるアドレスを叩くだけで，そのままそのバイトをポートに反映させることができます。

具体的に説明しましょう。このボードの，デフォルトでアドレスは$EC0000～$EC0007に設定されています。$EC0006（PPI1のコントロールレジスタのアドレス）に$80と最初に書けば，$EC0000にバイトデータをなにか書くだけで，そのままそのデータがパラレルポートに反映されることになるわけです。

出力はラッチされているので，次に同じ値を書き込むまで，同じ値がポートに出力され続けます。もっとも，コントロールレジスタにコントロールコードを書き込むことは，このICのリセットにもつながるので，すべて0にリセットされてしまいますが。

8255はMC6821と違い，ハンドシェイクがない分，このように簡単に使用できるのです。

とりあえずなにかつなごう

とりあえずパラレルでなにかつないでみましょう。最初にボードを受け取ったときにも悩んだのですが，意外と用途を思いつかないものです。

まず，最初にマイクロジョイスティックなるものを思いつきました。

要は，NEO・GEOや，メガドライブやPCエンジンなどの，ジョイスティック情報を横取りし，それをデータとして管理して，リプレイデータとして保存するというものです。ゲームが下手なあなたも，うまい人からアドバイスが受けられます……。

おお，いける！　と思ったけど，ちょっと時間がかかりそうなので今月は却下（特にソフトがねぇ，今月は忙しいのよ）。

次に考えたのは，ジョイスティック情報を管理することを前提にして，X68000を媒介にし電話回線で対戦ゲームをするというもの。お互いに，同じゲーム機とゲームを持っていて，乱数情報がほとんどないゲームでなくてはいけませんが，うまくいけば遊べそうです。餓狼伝説スペシャルや侍スピリッツが離れていても善ちゃんと対戦できるかなぁなんて。モデム通して電話回線使って，市内なら，1時間遊んでも，200円だもんね。50円のゲーセンの4回分だけでもとが取れちゃう。

でもソフト開発にものすごく時間がかかりそうなのでとりあえずこれも今月は却下。いずれやってみましょう。

そして，IDEインタフェイス。そろそろSCSIのID貧乏なんですよね。記憶に間違いがなければ，できないことはないとは思われます。なかなかいいと思うけどデバイスドライバがね。私に作れるかなぁ？

ということで，これは，誰かデバイスドライバを作ってくれるという有志がいたら，ハードウェアは私が作りましょう。ソフトは俺がやるぜ！　っていう方がひとりいたら試みてみることにします。

どれにしても，ものすごく時間がかかりそうです。48個ものLEDをトランジスタアレイにつなげてピカピカさせてみてもねぇ……と思いつつ，常用の部品箱を開けてみました。

大昔に買ったセラミックパッケージの綺麗な石。AD558という8ビットD/Aコンバータなのですが（要はPCM）このIC高かったんだよな。もっと安いはずなんだけどなぁ。500円ぐらいで売っていても不思議じゃないんだけど……なんで金メッキのセラ

ミックパッケージなんて買っちゃったんだろ。

このまま眠らせておくのももったいない。そんなわけで，簡単なPCM音源を作ってみることにしました。

AD558

「だいたい，AD558のマニュアルがないクセに，製作を始めるほうが間違ってるよなぁ，規格表しかないなんて……終わってる」

それでも時間もお金もないのでセコセコと接続。規格表を見る限り，それほど難しそうでもないICのようだったので使ってみることにしました。

この記事はPPIボードのレビューであって本筋のローテク実験室ではないので，回路の詳細はちょっと割愛させてもらいます。

今回の回路では48ビットもあるI/Oのうち8ビットしか使用していません。データは出力のみなので，基板のランドにはマニュアルの指示どおり，LS541をひとつ接続し，プルダウン集合抵抗を接続することにします。

すでに基板を改造してしまい，59，60ピンに電源を通してしまいました。X68000が拡張スロットに供給する電源は，1スロットあたり5V(Vcc1)で600mA程度です。このPPIボード自身がバッファ未装着時に120mA以下を消費するので，外部に供給できる電力は5V，480mA弱でしょう。これにケーブル自身の抵抗なども引かれてしまいますから，実際には安定して利用できるのは400mAぐらいでしょうか。

規格表を見るとAD558は75W利用するようなので，電流に直すと15mA使用されることがわかります。PPI基板上にバッファとしてLS541がひとつあり，定格では54mA，D/Aボード上に装着されたLS245は最大定格95mAです。あとはコレクタ接地されたトランジスタがいくつ電流が必要なのかちゃんと計算しないとわかりませんが，RLに1kΩを使用していることから，50mA程度でしょう。まあ，480mAにはほど遠いようですから大丈夫そうです。使用電流が気になるようならば，74LSを74HCに取り換えるのが吉というものですか。

さてこのパラレルインをPPIボードのPaにつなぎ，8ビット出力モードを利用して，じかにバス直結し，PCMボードを駆動させました。プログラム上の表記は簡単で，まずPPI1のコントロールレジスタ($EC0006)に$80を設定し(モード0，PA，PB，PCすべて出力)，あとはひたすらPAが直結している$EC0000に8ビットデータを叩くだけです。

このPCM-ICのデータ形式は，8ビットの符号なしPCMデータであり，PCM8の8ビットPCMデータの形式と違うものですから注意してください。$00が0V，$FFが2.55Vと，とてもわかりやすいものです。つまり，Cで表記するとしたら，unsigned charってところですかね。$80が真ん中あたりなので，注意してください。

実際のデータ出力プログラムはCのライブラリとして，リスト1に載せてあります。余談ですが，PCM8のPCMは符号つき8ビットPCM，すなわちCでいえばchar，FM TOWNSやメガCDのPCMはMSBが符号ビットであるPCMですから，これもまた同じデータをそのまま再生するにはデコーダが必要です。

これらのプログラム表記はわかりやすく，アセンブラ，Cの両方でPPIボードのマニュアルに記載されています。8255のそのもののマニュアルはないので，詳しい使い方はできませんが，私が今回解説したモード0だけで，十分事は足りると思われます。

AD558のデータ出力はトランジスタでコレクタ接地していますが，出力に直列で接続してある3kΩの抵抗は，半固定抵抗にしておいたほうがよいかもしれません。

さらに記載しておくとすると，この回路には，量子化ノイズ除去フィルタをつけていないので，量子化ノイズがバリバリ乗っているはずです。風の噂では，最近，X680x0のPCMを改造して，コンデンサを取り換えてローパスフィルタの特性を変え，量子化ノイズを鳴らして遊んでいる人がいるそうなので，時代としては面白いかもしれませんね。

どーしてもフィルタをつけたい人は，出力の+と−の間に適当なコンデンサを入れるとよいでしょう。これで簡単な一次フィルタになります。これならば，コンデンサのインピーダンスは，

$$Z = 1/(2\pi fC)$$

なので適当に決めてください。ここで細かく説明すると，何ページになるかわからないので割愛することにします。

AD558の記事ではないので詳しく書く気はありませんが，まぁ，こういう簡易的なものでもちょっとしたD/Aがあるのは面白いですねぇ。

真面目にアクティブ3次フィルタでもつければ，かなりよい音は鳴るでしょう。音源ソースが手に入らないという人はリスト2〜4を実行してみてください。それぞれ，短形波，サイン波，三角波の音が発生するはずです。周波数は機種によって変化します。

図1　AD558ボードの製作

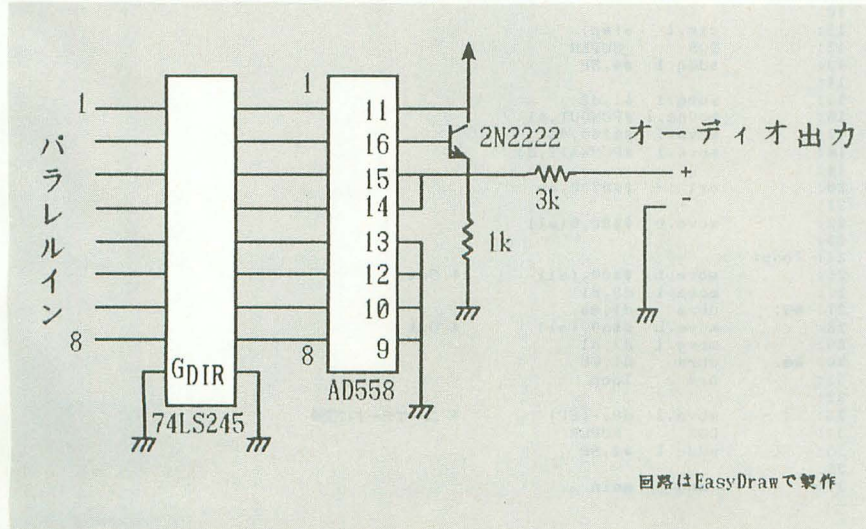



図2

8255のモード0での設定方法

ビット	7	6	5	4	3	2	1	0
値	1	0	0	PA	PCh	0	PB	PCl

PA：PA0〜PA7までの転送モード
1の時入力
0の時出力

PB：PB0〜PB7までの転送モード
1の時入力
0の時出力

PCh：PC4〜PA7までの転送モード
1の時入力
0の時出力

PCl：PA0〜PA3までの転送モード
1の時入力
0の時出力

全般を通じて

最初のほうにいったとおり，マニュアルは非常に親切で，ハードウェア設計の初心者にもかなり取っつきやすくできています。

高校や大学でデジタル回路を習い，いつかX680x0のオプションを自分の力でなにか作りたいという方には非常によいテキストになると思います。

なによりも物を語っているといえるその回路図，GALソースなども記載されていて，すべてを把握するのによくできた構造になっているといえるでしょう。

難をいえば，バッファのDIRを8255のモードにあわせて切り換えられないという構造ですか。回路図を熟読したわけではないのでわかりませんが，改造によってできないことはないはずなので，いずれローテクでなにかに使いたいときに利用してみようかと思います。同じものを作るよりもあるものを買ったほうが安い場合が多いですからね。

この回路を眺めて，ボードを利用していたときにふとX68030 Inside/Outの桒野さんの後書きを思い出しました。

なるほど。ソフトだけでなく，ハード作成も楽しむことができるパソコンは，X680x0が最後ではないだろうかと。

年々ブラックボックス化しているパーソナルコンピュータ。確かにX680x0のなかも簡単ではないですが，PC-9821のようにほとんどカスタムチップ化してしまったマシンに比べ，どこでなにをしているかわかるというのは，ハードを設計している者にも，これからハードを学びたい者にも，非常によいテキストになります（別にPC-9821が悪いといってるわけでは決してない）。

私は趣味で工学系の本をよく読みます。もちろん，InsideX68000, OutsideX68000のような個別のコンピュータの解析書（?）らしきものにもよく目を通します。1冊読むごとに，たいていなにか知識を得ることができますが，もっとも多くの知識を与えてくれるものは「いま現在そこで動いている」ハードウェアの回路図なのです。

Inside/Outside X68030は買いか?

「X68030ユーザーでプログラム，もしくはハードウェアをゴリゴリいじりたい人は買いだろうけど，僕はX68000ユーザーだからいらないかな？」そう思う人はたくさんいるに違いない。ところで，ちゃんと本屋さんで立ち読みしてみたかな？

今回，メガディスプレイ関係のページでいろいろ書いたけど，実は，X680x0シリーズのCRTCがつかめたのもすべてIn/Outのお蔭。そう。実は，INSIDE X68000，OUTSIDE X68000で載せ切れなかったことも，ちゃんと書いてあるんだな。これが。

この本の内容で非X68030ユーザーの私が興味深かったことといえば，CRTC関連と，68881/2の違い（英語マニュアルばっかでわからなかったんだよな），XVIから増設されたシステムポート，XVI以上から変わった拡張スロットのタイミングなどかな。

とりあえず，X68030ユーザーじゃなくても，X680x0シリーズをいぢる人には必要な1冊かもしれないよ。

リスト1

```
 1:             .include        doscall.mac
 2:
 3:             .xdef           _AD558
 4:
 5: PCMOUT      equ             $00ec0000
 6: PCMWAIT     equ             $00000080
 7:
 8:             .offset 4
 9: PCMADR      ds.l            1
10: PCMLEN      ds.l            1
11:
12:             .text
13:
14: _AD558:
15:             move.l  PCMADR(a7),a2
16:             move.l  PCMLEN(a7),d2
17:
18:             clr.l   -(sp)
19:             DOS             _SUPER              * 割り込み禁止するためにSuper
20:             addq.l  #4,SP
21:
22:             subq.l  #1,d2
23:             movea.l #PCMOUT,a1
24:             move.b  #$80,6(a1)                  * PPI ボード設定(Pa=out,Pb=out,Pc=o
ut)
25:
26:             ori             #$0700,sr           * 割り込み禁止(割り込みが入って音程が
27:                                                 * 変わらないように)
28:
29: loop:
30:             move.b  (a2)+,(a1)                  * Out
31:             move.l  #PCMWAIT,d1
32: @@:         dbra    d1,@b
33:             subq.l  #$1,d2
34:             cmpi.l  #$0,d2
35:             bge             loop
36:
37:             move.l  d0,-(SP)                    * ユーザモードに復帰
38:             DOS             _SUPER
39:             addq.l  #4,SP
40:             rts                                 * 戻る
41:
42:             .end    _AD558
```

リスト2

```
 1:             .include        doscall.mac
 2:             .include        iocscall.mac
 3:
 4: PCMOUT      equ     $00ec0000
 5: PCMWAIT     equ     $00000300
 6:
 7:             .text
 8:
 9: main:
10:
11:             clr.l   -(sp)
12:             DOS     _SUPER
13:             addq.l  #4,SP
14:
15:             subq.l  #1,d2
16:             movea.l #PCMOUT,a1
17:             move.l  #$100,d2
18:             move.l  #PCMWAIT,d3
19:
20:             ori     #$0700,sr
21:
22:             move.b  #$80,6(a1)
23:
24: loop:
25:             move.b  #$00,(a1)           * Out
26:             move.l  d3,d1
27: @@:         dbra    d1,@b
28:             move.b  #$a0,(a1)           * Out
29:             move.l  d3,d1
30: @@:         dbra    d1,@b
31:             bra     loop
32:
33:             move.l  d0,-(SP)            * ユーザモードに復帰
34:             DOS     _SUPER
35:             addq.l  #4,SP
36:
37:             .end    main
```

X68000/XVIの回路図は，OUTSIDE X68000に，X68030の回路図はX68030 Inside/outに記載されています。全回路図で，ものによってはPALの等価回路まで書かれています。

さすがにカスタムチップの中身までは書かれていませんが，ある程度周りのハードウェアから想像できますし，筆者の桒野氏の尽力で解析されたデータもまとめられています。

多少特殊な形とはいえ，自分がいま使っているパーソナルコンピュータ（ワークステーションですか？）の回路図が公開され，ユーザーが自由にいじることができるようなマシンは，いったい現在世界中にいくつあるでしょうか？

私の願いは，パソコンがもっと一般の人たちに使えるようになってくれることです。しかしこれは，いま私がいったことと矛盾しているかもしれません。ブラックボックス化することによって，初心者がわかりやすくなるのならば，マニアックともいえるハード解析者たちは，一歩譲らなくてはならないのでしょうか？

本音をいえばそうは思いたくないのが事実です。パーソナルなコンピュータなのですから，自分でなにを改造してもよいはず。保証がなくなってしまうとはいえ，それでも自分なりに改造してしまう楽しみは失われてはいないはずです。

そう思ってローテク実験室を始めたのです。内蔵されているICは純国産ではないにしろ，完全国産アーキテクチャのマシンが，いま日本から徐々に消えつつあります。用途を限定した最近の高性能マシンから見れば，X680x0は少々劣ってしまうかもしれません。また，ゲーム機能では専用機に勝てないでしょう。しかしながら，輝きはまだ失ってはいないはずです。散るならば美しく。散らぬならば根を深く。私は未来のために，身を奉げたい奉仕したいと思うのです。たとえ私が力尽きたとしても，きっと誰かが遺志を継いでくれるはず。そう信じています。

＊　＊　＊

製品紹介からは多少話がすぎてしまいましたが，個人的にいえば，拡張スロットが10個ぐらいあれば2枚ぐらいいつもつけていたいかな，なんて思うボードです。

そういえば，サードパーティからも増設拡張スロットの広告が掲載されていましたが，いつになったら発売されるのでしょうか……。

リスト3

```
 1:          .include        doscall.mac
 2:          .include        iocscall.mac
 3:
 4: PCMOUT   equ     $00ec0000
 5: PCMWAIT  equ     $0000000f
 6:
 7:          .text
 8:
 9: main:
10:
11:          clr.l   -(sp)
12:          DOS     _SUPER
13:          addq.l  #4,SP
14:
15:          subq.l  #1,d2
16:          movea.l #PCMOUT,a1
17:          move.l  #$100,d2
18:          move.l  #pcmdata,a0
19:          clr.l   d0
20:
21:          ori     #$0700,sr
22:
23:          move.b  #$80,6(a1)
24:
25: loop:
26:          andi.l  #$ff,d0
27:          move.b  (a0,d0.l),(a1)   * Out
28:          move.l  #PCMWAIT,d1
29: @@:      dbra    d1,@b
30:          addq.l  #1,d0
31:          bra     loop
32:
33:          move.l  d0,-(SP)         * ユーザモードに復帰
34:          DOS     _SUPER
35:          addq.l  #4,SP
36:
37:          .data
38:
39: pcmdata:
40:          dc.b    $80,$83,$86,$89,$8c,$8f,$92,$95,$99,$9c,$9f,$a2
41:          dc.b    $a5,$a8,$ab,$ae,$b1,$b4,$b6,$b9,$bc,$bf,$c2,$c4
42:          dc.b    $c7,$c9,$cc,$cf,$d1,$d3,$d6,$d8,$da,$dc,$df,$e1
43:          dc.b    $e3,$e5,$e7,$e8,$ea,$ec,$ee,$ef,$f1,$f2,$f3,$f5
44:          dc.b    $f6,$f7,$f8,$f9,$fa,$fb,$fc,$fd,$fd,$fe,$fe,$ff
45:          dc.b    $ff,$ff,$ff,$ff,$ff,$ff,$ff,$ff,$ff,$fe,$fe,$fd
46:          dc.b    $fd,$fc,$fb,$fb,$fa,$f9,$f8,$f7,$f5,$f4,$f3,$f1
47:          dc.b    $f0,$ee,$ed,$eb,$e9,$e8,$e6,$e4,$e2,$e0,$de,$db
48:          dc.b    $d9,$d7,$d5,$d2,$d0,$cd,$cb,$c8,$c6,$c3,$c0,$bd
49:          dc.b    $bb,$b8,$b5,$b2,$af,$ac,$a9,$a6,$a3,$a0,$9d,$9a
50:          dc.b    $97,$94,$91,$8e,$8b,$87,$84,$81,$7e,$7b,$78,$74
51:          dc.b    $71,$6e,$6b,$68,$65,$62,$5f,$5c,$59,$56,$53,$50
52:          dc.b    $4d,$4a,$47,$44,$42,$3f,$3c,$39,$37,$34,$32,$2f
53:          dc.b    $2d,$2a,$28,$26,$24,$21,$1f,$1d,$1b,$19,$17,$16
54:          dc.b    $14,$12,$11,$f,$e,$c,$b,$a,$8,$7,$6,$5,$4,$4,$3
55:          dc.b    $2,$2,$1,$1,$0,$0,$0,$0,$0,$0,$0,$0,$0,$0,$1,$1
56:          dc.b    $2,$2,$3,$4,$5,$6,$7,$8,$9,$a,$c,$d,$e,$10,$11
57:          dc.b    $13,$15,$17,$18,$1a,$1c,$1e,$20,$23,$25,$27,$29
58:          dc.b    $2c,$2e,$30,$33,$36,$38,$3b,$3d,$40,$43,$46,$49
59:          dc.b    $4b,$4e,$51,$54,$57,$5a,$5d,$60,$63,$66,$6a,$6d
60:          dc.b    $70,$73,$76,$79,$7c,$7f
61: pcmlen:  dc.b    $7f,$7f,$7f      * DUMMY
62:
63:          .end    main
```

リスト4

```
 1:          .include        doscall.mac
 2:          .include        iocscall.mac
 3:
 4: PCMOUT   equ     $00ec0000
 5: PCMWAIT  equ     $00000003
 6:
 7:          .text
 8:
 9: main:
10:
11:          clr.l   -(sp)
12:          DOS     _SUPER
13:          addq.l  #4,SP
14:
15:          movea.l #PCMOUT,a0
16:          move.b  #80,6(a0)        * ppi out
17:
18:          clr.l   d1               * インクリメンタ
19:
20: loop:
21:          andi.l  #$ff,d1
22:          move.b  d1,(a0)          * Out
23:          move.l  #PCMWAIT,d0
24: @@:      dbra    d0,@b
25:          addq.l  #$1,d1
26:          bra     loop
27:
28:          move.l  d0,-(SP)         * ユーザモードに復帰
29:          DOS     _SUPER
30:          addq.l  #4,SP
31:
32:          .end    main
```

パーソナルな「安心」

無停電電源装置BX3

Kioi Makoto 紀尾井 誠

本当はとっても必要なものなのに，誰も持っていません。あったからといって普段はなんの役にも立ちません。しかし，もしものときには素晴らしい働きをする……そんな周辺機器です。

瞬断0.2秒

この春，東京都を含め関東主要部を襲った大規模な停電事故。その時間わずかに0.2秒間（もうちょっと長かったのではないかという気もするが）。しかし，コンピュータにとっては致命的な時間だ。多くの人が被害にあったことと思われる。

編集部ではマシンで本体がリセットされて編集中のテキストが飛び，ついでにRAMディスクが飛んだ程度ですんだものの，被害が皆無というわけではなかった。

瞬断。

こういうことも起こることがあるとは聞いていたが，実際に被害にあったのは初めてだった。たった0.2秒にも耐えられないパソコンの脆さを痛感させられる出来事だった。こんなとき記憶の底から浮かび上がるのが無停電電源装置だ。

究極の贅沢かもしれない

無停電電源装置。説明するまでもないと思うが，これは停電などが発生したときに代わりにバッテリーから電源を供給してくれるような装置のことである。

世間を見ていると，いわゆるセキュリティというヤツにどうも関心が低いようだ。ハードディスクが1Gバイトあろうが，アクセラレータが爆速だろうが，電源がなければパソコンはただの箱にすぎない。性能追求というのもひとつの道ではあるが，ともすればただの「お子様趣味」になりがちである。パソコンの利用形態が高度になるほど，セキュリティ追求というのは重要であり，玄人の領域になってくる。車を買って，まずエンジンやマフラーをいじる奴とブレーキや足回りを固める奴とでは走りの次元が違ってくるものだ。

で，最近の無停電電源装置というものを調べてみると，かなりパーソナルユース指向のものも現れているようだ。価格で4万円くらいだから個人でも手が出ないというものではなくなった。その一方でパーソナルユースだって，どうしても失敗できない作業とか消えては困るデータを扱うことはある。ハードディスクの書き込み中に停電なんてことになったらちょっと怖い。

今回試用したのはオムロンのBX3。2つのユニットに分かれたタイプで，縦置き，横置き積み重ねとレイアウトは自在になっている。

BX3は最大300VA/180Wまでの負荷に対して動作する。パソコンやハードディスクなどの定格を合計して180W以内ならいいわけだ。BX3はバックアップ電源用タップを2個備えているので，電源の切れては困る機器をバックアップ型タップにつなぐ。X68000本体のサービスコンセントを使えば，X68000本体とほかに2つ程度の機器を接続できることになる。もちろんマルチタップを使えばもっと接続機器を増やせるのだが，そういう用途にはもっと大型の無停電電源装置を使うべきだろう。

さらに，通常の非バックアップ型タップを1個備えているので電源が切れても大丈夫なものはそちらにつなぐこともできる。

バックアップできる時間は最大定格時に5分。これは消費電力によって変わるのでX68000本体のみのバックアップなら定格の24%の負荷だから，30分弱もつ計算になる。普通に使うには十分な時間であろう。

一晩かけてレンダリングを行って……という場合の停電には対処できない可能性があるが，警報ブザーで目を醒ます自信があればこれも大丈夫だろう。BX3は停電やバッテリー切れ，オーバーロード時にLED表示とともにアラームを発する。さすがに非常用の装置だけあって警報面も怠りない。

問題はその間になにをすべきかだが，

●ブレーカが落ちた場合

ブレーカを上げる

●停電の場合

編集中のファイルをセーブして終了する。またはRAMディスクを保存する

ということになる。瞬断の場合ならなにもしなくても大丈夫だ。たった5分や10分だが，最近ではよほどのことがない限り，停電といっても数分で回復してしまうものだ。

実地テスト

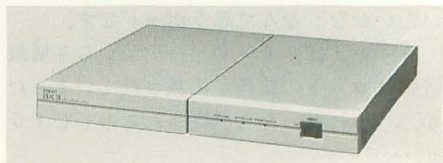
それでは実測してみよう。使用機器はX68030（HD80Mバイト）＋CU21HDだ。

フル充電で電源装置のコンセントを抜くと「ピ，ピ，ピ」とアラームが鳴り始める。かなり大きな音だ。9分50秒後，アラームが「ピー」という連続音に変わる。さらに1分後，ついに電源が切れる。

これでほぼ11分間電源を供給していたことになる。パソコンを目の前にしていれば，これはかなり長い時間である。

X68030本体の定格が38W，CU21HDはなんと135W。ハードディスクの分を入れるとちょうど定格ぎりぎりのところになる。これで10分以上もったのだから普通に使うには十分な性能と思われる。

どうしても不足というなら上位機のBX5（出力500VA/300W）というものもある。出力が120W（67%）も増えるが値段は4,200円しか違わないので，こちらのほうがおすすめかも（外形はBX3と同じ）。

あとはハードディスクのミラーリング機能さえあれば無敵の不沈パソコンなのだが，すでに矢でも鉄砲でも持ってこいという感じ。ああ，停電が待ち遠しい……。

オムロン　BX3　42,800円（税別）

BACK ISSUES バックナンバー案内

ここには1993年6月号から1994年5月号までをご紹介しました。現在1933年9～12月号，1994年1～5月号の在庫がございます。バックナンバーはお近くの書店にご注文ください。定期購読の申し込み方法は160ページを参照してください。

1993

6月号（品切れ）

創刊11周年特別企画　確率遊技シミュレーション

連載　DōGA CGアニメーション講座/こちらシステムX探偵事務所
響子 in CGわ～るど/ショートプロ/大人のためのX68000
ハード工作/吾輩はX68000である/Computer Music入門

●新製品紹介　SC-55mkII
LIVE in '93　ストリートファイターIIより 春麗のテーマ/BAY YARD/LOVE&CHAIN
THE SOFTOUCH　餓狼伝説/信長の野望・覇王伝 他
全機種共通システム　REVERSI

7月号（品切れ）

特集　席巻するローテク文明

連載　DōGA CGアニメーション講座/こちらシステムX探偵事務所
響子 in CGわ～るど/ショートプロ/マシン語プログラミング
ハード工作/吾輩はX68000である/Computer Music入門

●新製品紹介　ドローイングバット33070&MATIER
LIVE in '93　Midnight Circle/今日の日はさようなら/赤い靴
THE SOFTOUCH　悪魔城ドラキュラ/リブルラブル/大航海時代II/銀河英雄伝説III/幻影都市/ヴェルスナーグ戦乱
全機種共通システム　MSX用S-OS "SWORD"

8月号（品切れ）

特集　C言語実践的入門

連載　DōGA CGアニメーション講座/こちらシステムX探偵事務所
響子 in CGわ～るど/Computer Music入門/大人のためのX68000
吾輩はX68000である/ショートプロ/ANOTHER CG WORLD

●特別企画　夏真っ盛り，アマチュアリズムのX68000
LIVE in '93　SPLASH WAVE
THE SOFTOUCH　悪魔城ドラキュラ/リブルラブル/餓狼伝説/ロボットコンストラクションR.C./Winning Post
全機種共通システム　MACINTO-C再掲載

9月号

特集　光学式磁気円盤MO

連載　DōGA CGアニメーション講座/こちらシステムX探偵事務所
響子 in CGわ～るど/ショートプロ/大人のためのX68000
ハード工作/Computer Music入門/ANOTHER CG WORLD

●新製品紹介　OS-9/X68030
LIVE in '93　ファイナルファンタジーVのテーマ/銀河鉄道999/アルスラーン戦記IIより 汗血公路/ちょうちょ
THE SOFTOUCH　悪魔城ドラキュラ/コットン/ダーク・オデッセイ 他
全機種共通システム　7並べ/SLANG再々掲載

10月号

特別企画　秋祭りPRO-68K

連載　ハードコア3D/Computer Music入門/マシン語プログラミング
DōGA CGアニメーション講座/こちらシステムX探偵事務所
響子 in CGわ～るど/ショートプロ/吾輩はX68000である

●特別付録　秋祭りPRO-68K（5"2HD）
●SCSIパックンTOWER JACK
LIVE in '93　未来予想図II/OutRunより　PASSING BREEZE
THE SOFTOUCH　コットン/The World of X68000/あにまーじゃんV3
全機種共通システム　シューティングゲームコアシステム作成法(4)

11月号

特集　ポリゴナイザSLASHの活用

連載　ハードコア3D/Computer Music入門/ファイル共有の実験と実践
こちらシステムX探偵事務所/目指せジョイスティックの星
響子 in CGわ～るど/ショートプロ/大人のためのX68000

●新製品紹介　Easydraw SX-68K
OS-9 Ultra C/Technical Tool Kit
LIVE in '93　渚のアデリーヌ/エロティカ・セブン
THE SOFTOUCH　ぷたさん/ダイアット・ヴァークス
全機種共通システム　S-OSで学ぶZ80マシン語講座(1)

12月号

特集　古今東西ゲーム議論

連載　ハードコア3D/マシン語プログラミング/響子 in CGわ～るど
DōGA CGアニメーション講座/こちらシステムX探偵事務所
ショートプロ/Computer Music入門/ファイル共有の実験と実践

●新製品紹介　MATIER ver.2.0
C Compiler PRO-68K ver.2.1 NEW KIT
LIVE in '93　クリスマス・イブ/星に願いを
THE SOFTOUCH　ネメシス'90改/項劉記/スーパーリアル麻雀PII&PIII
全機種共通システム　エディタアセンブラREDA再掲載

1994

1月号

特集　Z-MUSICシステムver.2.0

連載　ハードコア3D/ゲーム作りのKNOW HOW/響子 in CGわ～るど
DōGA CGアニメーション講座/こちらシステムX探偵事務所
ショートプロ/Computer Music入門/ファイル共有の実験と実践

●特別企画　ANOTHER CG WORLD in Hong Kong
LIVE in '94　LAST WAVE/スターウォーズ/明日への扉/夢路より 他
THE SOFTOUCH　ストリートファイターIIダッシュ/餓狼伝説2/ドラゴンバスター/X68000傑作ゲーム選
全機種共通システム　S-OSで学ぶZ80マシン語講座(2)

2月号

特集　X-BASICとグラフィック

連載　ハードコア3D/ワンチップIC/響子 in CGわ～るど
DōGA CGアニメーション講座/こちらシステムX探偵事務所
ショートプロ/Computer Music入門/ANOTHER CG WORLD

●新製品紹介　ハイパーピクセルワークス
LIVE in '94　ランス3/新宿駅，巣鴨駅の発車メロディ/ピコー・ソング
THE SOFTOUCH　キーパー/マッドストーカーX68/餓狼伝説2 他
全機種共通システム　S-OSで学ぶZ80マシン語講座(3)
YGCSver.0.20リファレンスマニュアル

3月号

特別企画　ひなまつりPRO-68K

連載　ハードコア3D/マシン語プログラミング/ゲーム作りのKNOW HOW
DōGA CGアニメーション講座/こちらシステムX探偵事務所
ショートプロ/響子 in CGわ～るど/ファイル共有の実験と実践

●特別付録　ひなまつりPRO-68K（5"2HD）
●新製品紹介　ビデオPC for X680x0
LIVE in '94　THEME FROM WINNING RUN/スターフォースアレンジ版
THE SOFTOUCH　卒業/マッドストーカーX68/B-FIELD! 他
全機種共通システム　S-OSで学ぶZ80マシン語講座(4)

4月号

特集　SX-WINDOWの活用

連載　ハードコア3D/こちらシステムX探偵事務所
DōGA CGアニメーション講座/響子 in CGわ～るど
ショートプロ/ローテク工作/ANOTHER CG WORLD

●決定！　1993年度GAME OF THE YEAR
●新製品紹介　ビデオ入力ユニットCZ-6VS1
LIVE in '94　宇宙戦艦ヤマト/プロジェクトA子
THE SOFTOUCH　ジオグラフシール/ぷよぷよ/レッスルエンジェルス2 他
全機種共通システム　S-OSで学ぶZ80マシン語講座(5)

5月号

特別企画　こいのぼりPRO-68K
第9回言わせてくれなくちゃだワ

連載　ハードコア3D/響子 in CGわ～るど/ショートプロ
DōGA CGアニメーション講座/ファイル共有の実験と実践
こちらシステムX探偵事務所/ANOTHER CG WORLD

●特別付録　こいのぼりPRO-68K（5"2HD）
●新製品紹介　WorkroomSX-68K/開発キットツール集
LIVE in '94　ロード/時間旅行
THE SOFTOUCH　大魔界村/アルゴスの戦士/ジオグラフシール 他

ローテク工作実験室　第2回

パワースイッチ連動コンセントの製作

Taki Yasushi 瀧　康史

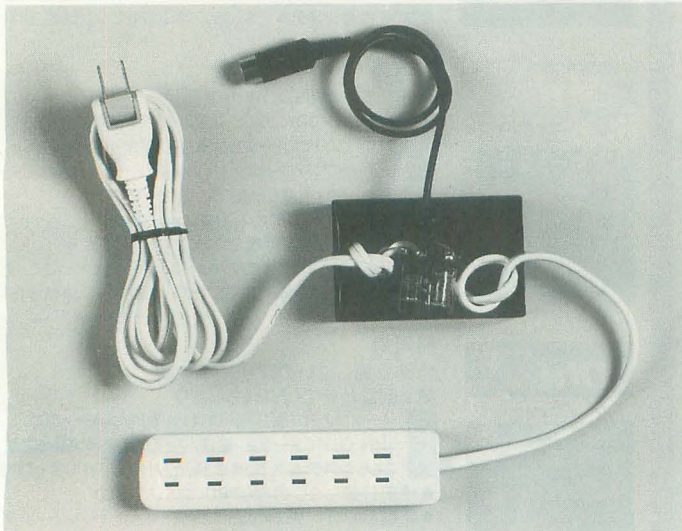
とっても簡単でそれなりに役に立つ，これぞローテク，ということで，X68000につなぐ連動電源コンセントを製作してみましょう。人によっては切実な問題になっているみたいですし。

あくまでもローテク

あくまでもローテク。

だから，難しいことはおいておき，生活の知恵のようなことをやってみました。ないとあるとでは随分違う，あれば生活が楽しくなる，快適になる，みたいな。生活といっても，コンピュータライフですけど。そっちのほうがネタがありますしね。

今月は，その生活の知恵で思いっきり攻めてみました。思いっきりローテク。だから回路も簡単，製作も簡単。すでに私が利用しているようなもので，人に見せたらわりあい好評だったもの。それはパワースイッチを連動したコンセントの作り方です。これなんかは切実にほしがっている人はいますよね。

モニタを制御するためのコントロールケーブルがあるためか，X680x0には残念ながら，本体の電源と連動してくれるサービスコンセントがありません。一応，本体と連動しないサービスコンセントはついていますけど，これ，結構，定格低そうですしね。タコ足なんて怖くってできやしません。つまり，ここにはモニタだけをつなげってことなのでしょうか。

ひと昔前ならこれはこれで困らなかったのですけど，最近はコンピュータと連動して電源をON/OFFしてほしいオプションが増えてきました。そうなると，NECパソコンみたいに本体と連動するコンセントがほしいところです。

特に最近の外付けハードディスク，PC-9801にあわせているのか，電源スイッチが背面にある場合が多いのです。マンハッタンシェイプの美しいX680x0と隣り合わせに，できるだけ見栄えよく置いても，電源スイッチのために自分の手を背面に通す隙間を考えておかなくっちゃいけません。困ったハードディスクだと，プッシュ式のID変更ロータリースイッチが隣にあって，電源スイッチと間違えて思わず押してしまいます。こうなるとせっかく綺麗に置いていても，引きずり出さなければならないのです。

PC-9801などのサービスコンセントは，本体電源と連動してON/OFFする（以下連動コンセント）ので，こういったときにまったく困りません。むしろ，電源スイッチが後ろにいくことで，スイッチのデザインを気にせず，汎用的なものを使えるのだから，コストダウンにも若干ながら手を貸していることでしょう。

ないものを作らなくちゃいけないのがX680x0ユーザーの宿命。要は自分のMOを連動させたいがため，作っただけの話です（ハードディスクは本体に2台も内蔵してるのさ)。回路は簡単，製作も簡単。大きな落とし穴がありますが，それも注意深く気をつければ大丈夫です。

たったこれだけ連動コンセント

部品説明

連動電源とほぼ同じような回路で，ソフトによりON/OFFをコントロールできる電源を作ることができます。こちらの利用は，主に，モデムなどのON/OFFや，イメージスキャナなどのON/OFF，ものによってはプリンタなどのON/OFFのコントロール，つまり，パソコンを使用するときに常には利用しないオプションの電源をコントロールするのに便利でしょう。

これらの方法は，ソフトを作らねばなりませんし，場合によりけりでいろいろ変わるので，要望が多くきたらソフトと一緒に作ることにしましょう。

部品表を見てください。部品はたったこれだけ。主な部品はリレーです。順を追ってなんのために必要なのか解説します。

1のダイオードは電子回路でよく使われるシリコンダイオードで，地方のパーツ屋さんにも置いてあるようなものです。このダイオードは回路中，コンピュータ側の保護のために使われます。流れる電流はわずかですので，適当なもので構いませんが，私は，1S1588を利用しました。先月，Wave Blasterの接続でフォトカプラの近くに利用したものと同じものです。

2のリレーはメーカー共通の型番が調べられないので型番記載はパス（リレーなどの共通型番が記載されている本があったら，誰か教えてください)。でもこれは，パーツ屋さんに行って，直流5Vでリレーが動き，交流200Vくらいの電圧がかけられるリレーをくださいといえば，まず間違いなく手に入ると思います。ついでにいえば，ここ

で，できる限り，DC5Vのほうは高インピーダンスで小電力で動き，AC100Vのほうは自分が利用したい周辺機器に見あったものにしてください。

延長ケーブルなどで15Aとか1500Wとか記載されているものがあるでしょう？ 15Aもあったらかなりタコ足してもいいよね？ 手もとにあるXVIは100W弱だし。

この記載は，使われているすべての部品のなかでいちばん定格の小さな部品のものなのです。したがって，いくらリレーの定格が15Aの大きなやつを購入しても，コンセントやケーブルなどでケチって，1Aなんてものを買ったら，それだけでその連動コンセントの定格は1Aになってしまいます。

わかってるとは思いますが，W（ワット：電力）＝A（アンペア：電流）×V（ボルト：電圧）ですからね。家庭用交流電源は，実効値100Vなので，15Aと1500Wは同じ大きさってことなんです。ものによって記載のしかたが違っているので注意してください。また，家庭用電源は実効値，100Vですが最大約140Vほどなので，リレーは約140V以上のものを買わなくてはならないのです。念のため。

リレーのケースが透明なタイプを購入された方ならすぐわかりますが，リレーとは電磁石によって物理的にスイッチのON/OFFをする単純な「器械」です。つまり，この場合5Vは電磁石のための電源なのです。

この5Vが一瞬で切れたらどうなるでしょうか？ 電磁石には残留磁気が残り，レンツの法則に従って逆起電力をひき起こします。すると，GND-VCC間に電圧が生じ，本体に負荷をかけます。そこで，これをショートするためにダイオードを挟むわけです。なくてもX680x0が耐えてくれればそれで動きますが，あれば事故防止になるってことはわかりますよね？ 20円ごときをケチってあなたのX680x0が壊れないことをお祈りしておきましょう。

3はコンセントです。3,4,5をまとめて説明しますと，これだけでAC電源コンセントの延長ケーブルが完成してしまうんですね。作るのが面倒くさければ，電気屋で15Aぐらいの延長ケーブルを買ってもよいのではないでしょうか？ さっきも少しふれたように，ちゃんとコードにも定格がありますから，気をつけて。少し考えれば当然ですが，140V,1A流せるコードと，140V，15A流せるコードでは，当然，太さが違います。コードだけじゃなく，コンセントそのものも違いますからね。

連動で動かしたい周辺機器がたくさんあるのならば，当然コンセントのメスコネクタがたくさんいります。その周辺機器最大定格の合計よりもすべてが大きくないと，これは故障の原因どころか，火事の原因にもなるわけですから，ご注意。

ジャンクで安いケーブルは，1Aぐらいしか流せないものがたくさんあります。だから，定格がちゃんと書いてないコードなんかは使わないように。夜中に漏電などによる火事が心配になって起きてしまうようなら，よいケーブルを買いましょう。もちろん，接続も間違えないようにね。

私は，メスコンセント，オスコンセント，ケーブル，リレー，すべて，15A流せるものにしておきました。

6はパソコンとの接続部分ですから，コードにはそれほど電流が流れません。ですからコードは1芯シールドにVCCとGNDを流すだけでよいでしょう。

しかし電源を引き出してくるケーブルには注意が必要です。あえて記載しませんでしたが，これは機種によって違うからです。やりたいことは，本体の電源と連動してくれる5VとVccが欲しいだけ。つまり電源が出ている拡張コネクタからは，どれからでも簡単に取ることができます。

まずいちばんメジャーなのは，ジョイスティック端子を使用するものです。ジョイスティック端子は，1P側も2P側もD-SUB9ピンのメスです。5ピンと9ピンにそれぞれ5VとGNDが出ています。

ときどき8ピンをGNDにする人がいますがこれは絶対にやめましょう。X680x0ではアタリ社仕様と互換を取るため，8ピンのアウトをGNDレベルに出力しているだ

部品表

1．ダイオード（IS1588）
2．リレー（5V-DCでリレーが動き，141Vの交流電源が流せるもの）
3．コンセント（オス）
4．AC100Vを流すケーブル
5．コンセント（メス）*n
6．各自のX680x0からVCCとGNDを取ってくるためのコネクタ，およびケーブル
7．リレーをつけるための基板
ちっちゃなものでよい
8．リレーを入れるための箱
透明プラスティックケースだと夜が楽しくなる

ストIIダッシュを安上がりに

ストリートファイターIIダッシュはCPSファイターMDが使えるという事実から，見当でメガドラ用の6ボタンパッドが使えるようにするアダプタを作ってみました。

見当といっても技術的裏づけがまったくないわけではありません。実はストIIが出た時点で，ある程度やり方が想像できていたんですよね。

以前，メガドライブ本体と，メガドライブのパッドを解析したとき，メガドライブのD-SUB9ピン（ジョイスティック端子）の仕様を調べてみました。

結果，3ボタンのパッドは74HC157がひとつ。6ボタンはSEGAカスタムチップで構成されていて解析できなくても，互換性がある程度あるわけですからインタフェイスに変化があるわけがありません。

実はメガドライブのD-SUB9ピンはX680x0のジョイスティック端子に非常に似ていて，X680x0のアタリ仕様ジョイスティックもメガドライブでボタン1個ならば使用することができます。X680x0との違いは3つ。それは，7．8．9ピンの機能です。

メガドライブのジョイスティックポートは7はアウト，8はグランド，9はインです。一方X680x0のジョイスティックポートは7がイン（ボタンA），8がアウト，9がグランド。この違いを吸収するように，延長ケーブルの途中をちぎって，うまくつなげてみました。

この予想どおりメガドライブの6BパッドでもストIIダッシュができるではないですか。若干違いがあり，ストIIダッシュではABCボタンをCABにしなくてはいけなませんが。

いろいろやってみましたが，チェルノブも，餓狼伝説1と2もできます。そういうわけで，なんとなく作ってしまったのが以下の配線表。

まぁほとんど適当に作ったら動いてしまったので，各自の責任のもとで作ってください。

メガドライブを持ってなくて，ゲームをするマシンは，X680x0だというユーザーは，セガのアーケードスティック6Bを買って中身を開けて，配線を入れ替えるのがよいのではないでしょうか？ あのスティックは重くて使用しやすいようですしね。

●変換ピン配置表

X680x0側	MDPad側
1	1
2	2
3	3
4	4
5	5
6	6
7	9
8	7
9	8

けで，実際はあくまでOUT端子です。

こういった背景でジョイスティック端子は電源を取るのに確かに向いています。しかし，マンハッタンシェイプの人ならジョイスティック2Pが後ろにあるので目立たなくてよいのですが，PROの人の場合表に出てしまい，いささか間抜けです。

それに，マンハッタンシェイプの人でも，ジョイスティックを利用しない人ならば，これでいいかもしれませんが，ゲーマーな私にはジョイスティックポートが埋まってしまうなんて耐えられません。友達を呼んで「あすか120%」で対戦できなくなってしまう……（レビューが書けん）。

なんといっても，連動コンセントは，パソコンを使用している最中，「いつでも」利用するものなのですから。

こんなことのために拡張スロットを使うなんて考えられませんし，マウスコネクタやキーボードコネクタみたいに，前にあるコネクタはさすがに使いたくないですよね？ そこで，PROの人，CompactXVI以降のユーザーにはまたもや申しわけないですが，立体視端子が狙い目なのです。これは滅多に使いませんし（私は立体視スコープつないでますけど）。ただ，ミニDIN6ピンなので，かなり探さないといけないでしょう。どうしてもなければ，ミニDIN8を買って真ん中をちぎるとよいのですが。

あとはほとんど電源を専用に取ることはできません。したがってなにかから電源を横取りしてくる必要があります。

ケーブルは当然自作。

電源を盗むならば盗みやすい端子を探すのがコツ。コンパクト以外のユーザーなら，専用テレビコントロール端子が狙い目です。DIN8ピンは割とコネクタはメジャーですし，コネクタ径が大きいのでTVコントロールの傍ら，ちょっと電源を拝借しやすいのです。

最近，モニタはナナオとかソニーのトリニトロン管をつなげたりしている人が多いので，そういう人にとって，この端子はまったく無用ですからねぇ（ナナオのモニタは輝度で電源をON/OFFする機構がついていたかな？）。

7はリレーをつけるためのケース。部品は2つとたいした数じゃないので，リレーが載ればそれで満足。基板をケチってもいいけどハンダづけに慣れていない人は買ったほうが楽なんじゃないかな？ 先週おすすめしたホットボンドなんかで途中を固めておいてもよいかもしれません。そしたら，ケースはいらなくなるし。でもちょっと技術が必要かな。

8はケース。リレーに透明なものを購入していれば，ケースを透明にするのが面白いでしょう。リレーのなかの電磁石の接続が丸見えだから。火花がバチ！ って飛んで夜は面白いぞ。

そういうのが嫌いな人は真っ黒な箱を買いましょうね。

制作手順

製作手順といっても，この程度の回路ですから，手順もなにもありません。あるがままに作ればよいのですが，最初にいった大きな落とし穴を避けるために注意が必要になります。

まずリレーのスイッチ側（X680x0側）と動作側（コンセント側）では電気回路はまったく別物ということです。

一方は直流5V程度，電流ミリアンペアの弱電。そしてもう一方は交流100Vでさらに大電流。この2つの電流がひとつの小さなIC？ に入るのですから，これには細心の注意を払わねばなりません。

想像してみれば容易だと思いますが，リレーのスイッチ側と動作側は絶対に接触しないように，テスターで念入りにチェックしてください。この2つが万が一にも接触した場合，確実にX680x0が壊れてしまうでしょう。

また，できあがったと思われたときでも，いきなり動かしてみず，スイッチ側のVccとGNDがショートしていないかチェックをし，コンセント側も2つがショートしていないか必ずチェックしてください。

ここまできちっとできたら，まずX680x0本体に連動電源のコネクタを差し込み，AC100V側は差さないでいます。リレーが透明なケースだったなら，X680x0の本体の電源がON/OFFされるごとに，リレーがちゃんと入るかどうか確認してください。

テスターでもリレーの動作側がちゃんとX680x0本体の電源に連動してスイッチが入るか確認します。

すべてを確認し終えたうえで，いったんX680x0の電源を落とし，連動電源のコンセントも差し込み，コンセントが連動してON/OFFされるか確認してみてください。

また，作成したコンセントは，多分ひとつではなく，3つぐらいのタップにするでしょうから，コンセントはX680x0本体からは取らないようにしてください。

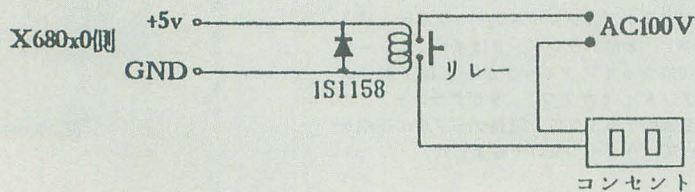
当然ながら，一歩間違えば危険な回路です。あなたの大切なX680x0が壊れてしまう程度ならまだ可愛いもので，最悪ケースでは，いったん動いたように見えても，必要以上の電流が流れて，漏電し，火事になってしまうおそれがあります。回路は確かに簡単なものですが，電気機器のアンペア計算，タコ足配線を計算できる中学校理科程度の電気回路の知識は必要だといえるでしょう。

壊してしまったなんとかしてくれ，といわれても，当然私にはなにもできません。すべてが自分の責任になると承知のうえで，製作するようにお願いします。

追記しておきますと，この回路でX680x0を壊してしまったとしたら，多分メインボード交換になります。X68030を買ってついでに040Turboでも買いましょう。修理代はおそらく20万円近くなると思うので。X68030を壊してしまったら，次のマシンでも待ったほうがいいのかな？

もっとも簡単，かつ，安全に回路を作成する方法は，市販されているコンセント延長ケーブルを利用することです。こうすることによって，余計な部分の危険性が減りますし，なにより楽です。ケーブルの途中を切り，その部分にリレーをつけ，ケースに入れるのが得策でしょう。

なお，連動コンセントというのは所詮，電源コンセントを，抜いた，差したで電源のON/OFFを行うことができるハードしか制御できません。できると「思われる」ものは，モニタ，HDD（ただしSCSIに限る），MIDI楽器，MOぐらいで，これらは商品によって異なります。MOでも，シャープのCZ-6MO1はガラス細工のMOといわれ

図1

ていて，どうやら確実に壊れるという噂が飛び交っています（いやすぎ）。

3.5インチMOの場合は致命傷にはなりませんが，もともとゴミやほこりにはデリケートな機器ですので気をつけましょう。ディスクが入ったままの状態で電源を切ると，隙間からほこりが入ってしまいます。必ずディスクを抜いてから電源を落としてください。

また，インクジェットプリンタなどは，電源OFF時に，必ずアルコールでノズルメンテナンスを行うので，これによって電源をON/OFFすると，故障の原因となること請合いでしょう。私の家でもIO-735Xは丁重に自分の手で電源を入れています。

最後に

くどいようですが，この記事を見て作るハードウェアの責任はすべて自分自身で取ってください。編集部に電話していただいても，私は常に編集部にいるわけではありませんし，むしろ，私は編集部には月に数回しか顔を出しません。当然，これらすべての内容の詳細を編集の人に話しているわけではないので，編集の人も相談されても困ってしまいます。

私に確実に届けたい場合は，封書などで，編集部内，私宛にお手紙をください。

生きた情報がほしい場合，パソコン通信上で全国ネットPCVAN内X1CLUBフォーラム7．ハードウェアにてサポートしています。

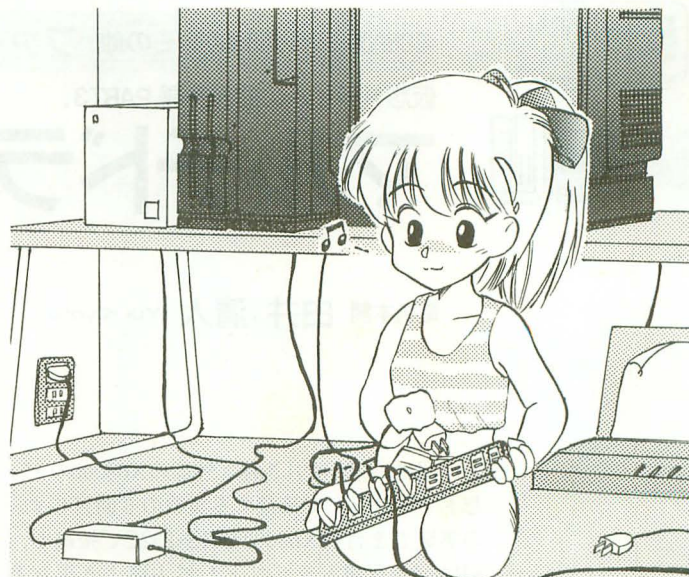
そういえば，4月の回路に友人から突っ込みが入ってきました。

「でも，あれじゃあオペアンプの規格表がないと作れないよね」

なるほど。そういえばフォトカプラは書いたのに，オペアンプは書き忘れてたなぁ。

モノシリックオペアンプの規格表にLF356は載ってますから，各自で買うようにしてください（わざわざ教えないところがミソ。だってパッケージによって違うもーん）。

まあ，オペアンプの規格表を買い惜しみするようでは，なにか問題が起きたときに自分で解決できませんってば（責任は自分で取るように……なんですから）。

今月は猛烈なローテクでしたがいかがでしたでしょうか。え？　リレーの何番ピンがなんの端子かわからない？　テスターを買って各自で調べましょうね。

さて，気が向いたら，来月もまたやります。いまモニタ2股器を作っていますが，できあがったら記事に仕上げてみましょう。　なにに使うかって？

そんなもの，必要だから作るに決まってるじゃないですか（必要じゃなくても作るけどさ）。

それでは今月はこの程度にしましょう。

麗しの青LED再び

ずいぶん前のことですが，連載とは別に本体のLEDを青くするという記事を書きました。わりあい好評なようで，通信をしていて他人のプロフィールを見たときに，「青LED取り替え済み」などと誇らしげに書いてあると，ちょっと嬉しいようなまずいような複雑な気持ちになります。

前回の記事を書いたあと，なんとか青いLEDがもっと明るくつかないだろうか？　といろいろ考えてみました。何度か実験をし，青LEDを破壊したりした挙句，無改造の緑並みに明るくする方法がありました。

LEDをもっとも簡単に明るくする方法は，かける電圧を上げることです。5V程度の電圧を当ててあげれば，無改造の緑並みに簡単に明るく輝きます。

しかし，当然のことですが，これは定格以上の利用方法であってLEDがまるで燃え尽きるかのように劣化も速くなります。実際に燃えるのではなく，長い時間5Vという電圧をかけると徐々に暗くなっていくのがわかります。

さて，5Vを青LEDにかけてみたところ，わずか数時間の連続使用で，徐々に暗くなり，次第に点灯しているのがわからないほど暗くなっていってしまいました。仮に劣化して使えなくなったにしても，10カ月ほどもつのならば，私は10カ月に一度くらいはX680x0を必ず開けているため問題はありません。しかし1日ももたないのならばこの電圧はだめです。

明るさと電圧はほぼ比例し，逆に寿命と明るさはこれでは反比例しています。

差し引きがちょうどよいくらいにするにはどうすればいいのでしょうか？　私は結局5Vに対してLEDとの間に200～300Ωぐらいの抵抗を挟み，（通常は500Ωぐらい）お茶を濁すことにしました。現在，取りつけてからすでに2カ月ほどたっていますが，LEDはまだまだ明るく輝いています。

けれども一度でも短時間5Vで青LEDをつけて見てしまったら最後。あの目映いほどに麗しい青は，あなたの目にも心にも焼きついて離れないでしょう。それでも実際問題，あの明るさではLEDの劣化は激しく使いものにはなりません。

ではどうすればいいのでしょうか？　どうすれば明るく照らすことができるのでしょうか？

そこで，ちょっとした実験をしてみました。本来ハードウェアで組むのが適当なのですが，LED基板のタイマランプを青にし，LEDにぶら下がった電圧降下用の抵抗を除去し，錫メッキ線でショートし（0Ω），定格以上の電圧をかけます。

このままタイマランプをつけると次第にLEDは燃え尽きてしまいます。そこでまるで蛍光燈のようにソフトでON/OFFを繰り返し，目につかないくらいの速さで点滅させます。こうすると0Ωのときの明るさよりは暗いのですが，最初についている抵抗をそのままつけているよりも明るくなります。そしてこのまま一晩中つけていましたが，燃え尽きる様子はありませんでした。

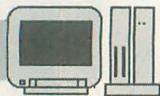
つまり，点灯したままでなければ，LEDの劣化はある程度防げることがわかりました。

今回，これをハードウェアで行う回路は机上では作りましたがまだ実践していません。しかし，X680x0のHD-BUSY，もしくはタイマランプをなんらかのフリーソフトのRAMディスクドライバなどでRAMDISKのアクセスランプに利用しているのならば，上記の実験と同じ理由により抵抗を除去することができます。

つまりアクセスランプは常に点灯しているわけではないからです。一瞬ならば電圧を大きくかけても，大丈夫……かどうかはわからないのですが，この場合，それが当てはまるらしく，私のXVIはもはや2カ月ほどこの状態で耐えています。

どうやらこの考え方は当たっているようで，実際，パルスなどを利用して定格以上の明るさを出す方法があるそうです。

以上，ほとんどヒントに近いものですが，お役に立つならば幸いです。

仮想ドライバの開発実験PART3.

マルチドライバ化への挑戦

電机本舗 由井 清人 Yui Kiyoto

今回は5月号でできあがったSRAMディスク用のドライバを汎用のディスクドライブに使えるようなマルチドライバとして作り直します。これで従機側のX68000のドライブすべてを使用することができます。

今回は，前回に引き続き外部のX68000のディスクを仮想ドライブとして接続する実験の3回目です。前回2回の実験で2台のX68000をRS-232Cで接続し，2台目のSRAMを仮想ドライブとして主機より読み書きしてみました。

今月は，このプログラムをさらに改良して，SRAMだけでなくフロッピーディスクやハードディスクを同様に制御する実験をします。

これにより，フロッピーやハードディスクを自由に仮想ドライブ化できるようになります。今回の実験では試験をしていませんが，MOやCD-ROMなどであれ，従機のX68000につながっているすべてのドライブを利用できるように設計しました。

実際のテストはフロッピーとSRAMディスクに対して行いました。ハードディスクはディレクトリ参照および読み取りは行っていますが仮想ドライブのハードディスクへの書き込みテストはしていません。これは，バグなどによりハードディスクのデータの破壊が怖かったからです。読者の方も動かすときはフロッピーないしRAMディスクに限定しておいてください。テストする場合はハードディスクのバックアップをきちんと取ったあとで，試してみてください。

マルチドライバ化の条件

これまでのおさらいになりますが，ここでブロック型デバイスドライバ（ディスク装置のことと思ってください）の内部構造に触れておきます。詳しくは当連載の4ないし5回目を参照してください。

ブロック型デバイスドライバは，BPB（BIOSパラメータブロック）テーブルと呼ばれるデータ領域を内部に持っています。

このデータ領域は，そのブロック型デバイスドライバの仕様を定義するものです。すなわち，ディスク容量，セクタ数，ファイルを管理するためのディレクトリ領域の大きさなどを所有します。

BPBテーブルはドライブ装置1個につきひとつ定義することができます。ですから，仮想ドライブのマルチ化はBPBテーブルを複数持つようにすれば解決できそうです。

前回作った仮想SRAMディスクドライバは，SRAMだけを前提にしていたので，このBPBテーブルをひとつだけ用意していました。また，BPBテーブルの内容もSRAMディスクなので，固定で十分に間に合いました。

今回は，このBPBテーブルの制御を変更して最大10組まで持てるように拡張しました。さらに，固定で持たせていたBPBテーブルの内容を従機の仮想ドライブに指定したディスク装置を実際に調査して，この情報を元にBPBテーブルの内容を作成するようにしました。

具体的なプログラム

文中で参照するリスト表を次に示します。

主機側のプログラムは前回，前々回に作った「d0.x」をベースに利用します。

「d0.x」は次の4本のプログラムより作ります。これらは，当連載第6回（3月号）に掲載されたファイルです（ただし，プログラムの修正情報が5月号に掲載されたので要注意）。

D0.S　主機側仮想デバイスドライバプログラム本体
D1.C　C言語で構築された内部処理プログラム
D3.C　通信の入出力ルーチン
cx.bat　コンパイルリンク用バッチ

今回は，これらのファイルに対して次の変更を加えて主機側のプログラムを作ります。

BPB領域の拡張

手始めに複数のドライブに対応するためにBPBテーブルの数を10組に拡張します。

「D0.S」をリスト1のように変更してください。

「D0.S」はアセンブラで記述されたデバイスドライバの本体プログラムです。ここにBPBテーブルがひとつ領域確保されています。

従来，87行目から95行目までの間の領域がBPBテーブルとして機能していました。ここには，これまでの仕様ですと，SRAM用のBPB情報が固定で設定されていたわけです。

これを変更後の87行目から105行目のようにします。

87行目から95行までというのは，最初のBPB領域です。97行目から105行目までは残りの9つのBPB領域です。最初のBPBテーブルだけ領域の確保の仕方が違うの

は，ほかのプログラムのラベル定義などを変更したくなかったためです。それ以上の意味はありません。プログラム的にはメモリ領域が数だけ確保されていれば問題なしです。

bpbtbl1という名で領域確保されていたBPBテーブルが変更後は10組に拡張されているのがわかるはずです。

同様に，_inittblはBPBテーブルのアドレスを管理している領域です。ここに，今回拡張されたBPBも含めて10組登録してあげればよいわけです。

Human68kはこの_inittblを参照し，ここに登録してあるBPBテーブルに従いディスク装置を認識します。

また，ここで用意した10組という数字に特に根拠はありません。とりあえずこれだけあれば十分であろうということから確保しました。

あとは，このBPBに仮想ドライブのディスク情報をきちんとセットしてあげれば，きちんとマルチドライブに対応して動作するようになります。

BPB領域への設定

このように確保されたBPBに対して，正しくディスク情報を設定する必要があります。この設定は主機に登録する仮想ドライブの数だけ行います。この機能を「D1.C」プログラムの中に組み込んでみましょう。

BPB情報をHuman68kへ提出する役目を果たす関数「dskini()」改造します。この関数は，従来はすでにSRAMディスク用に定義されていたBPBテーブルをHuman68kに受け渡しているだけでした。これを自ら従機のドライブのBPB情報を調べて，BPBテーブルへ格納，そしてこのアドレスをHuman68kに渡すようにします。このBPB情報の調査そのものは従機のほうで行いますので，ここでは変更方法と従機との通信制御を説明します。

まず，変更情報をリスト2に示します。

修正が終わったならば，cx.batファイルを実行してコンパイルリンクしておいてください。

変更後のプログラムそのものは難しいものではありません。従機側のプログラムと通信を行いBPB情報を送ってもらい，BPBテーブルへ格納しているだけです。

このdskini()という関数はデバイスドライバの初期化プログラムです。

Human68kが立ち上がるときにデバイスドライバの登録が行われます。このときに，この関数が呼ばれます。この一連の動きは当連載の4回目（1993年12月号）と5回目（1994年1月号）で説明しています。不明の方はこちらを参照してください。

dskini()は，リクエストヘッダと呼ばれるHuman68kからの機能要求に従い，BPBテーブルにディスクの仕様を設定し，Human68kに送り返しています。このあたりの処理は具体的には494行目から499行目までの場所で行っています。ここでリクエストヘッダの構造体に対して，エラーコード，ドライブの数，BPBテーブルの管理メモリを格納してHuman68kへ知らせています。

BPB情報の格納は485行目から511行目の間でループさせています。ここで仮想ドライブの数だけ回し，各ドライブBPB情報を取得してBPBテーブルに登録しています。

表1　BPBの内容（12バイト）

サイズ	内　容
1W	1セクタあたりのバイト数
1B	1クラスタあたりのセクタ数
1B	FAT領域の個数
1W	予約領域のセクタ数
1W	ルートディレクトリに入るエントリ数
1W	全領域のセクタ数
1B	メディアバイト
1B	1個のFAT領域に使用するセクタ数

表2　参考：DPBの内容（84バイト）

サイズ	内　容
1B	装置番号　0＝A：　1＝B：
1B	ドライバ内でのユニット番号
1W	1セクタあたりのバイト数
1B	1クラスタあたりのセクタ数－1
1B	先頭クラスタのセクタ番号
1W	FATの先頭のセクタ番号
1B	FAT領域の個数
1B	1個のFAT領域に使用するセクタ数
1W	ルートディレクトリに入るエントリ数
1W	データ部の先頭セクタ番号
1W	総クラスタ数＋1
1W	ルートディレクトリの先頭セクタ番号
1L	デバイスドライバへのポインタ
1B	メディアバイト
1B	DPB使用フラグ（－1でアクセスなし）
1L	次のDPBへのポインタ
1W	カレントディレクトリのクラスタ番号
64B	カレントディレクトリの文字バッファ

リスト1

```
------------- 変更前 ---------------
87:     bpbtbl:
88:             dc.w    1024
89:             dc.b    1
90:     bpbftc: dc.b    1
91:             dc.w    1
92:     bpbdrc: dc.w    32
93:     _rdmaxr:        dc.w    $10
94:     bpbfid: dc.b    $f9
95:     bpbfsz: dc.b    1
96:
97:     _inittbl:
98:             dc.l    bpbtbl
------------- 変更後 ---------------
87:     bpbtbl1:
88:             dc.w    0
89:             dc.b    0
90:             dc.b    0
91:             dc.w    0
92:             dc.w    0
93:     _rdmaxr:        dc.w    0
94:             dc.b    0
95:             dc.b    0
96:
97:     bpbtbl2:        dc.b    0,0,0,0,0,0,0,0,0,0,0,0
98:     bpbtbl3:        dc.b    0,0,0,0,0,0,0,0,0,0,0,0
99:     bpbtbl4:        dc.b    0,0,0,0,0,0,0,0,0,0,0,0
100:    bpbtbl5:        dc.b    0,0,0,0,0,0,0,0,0,0,0,0
101:    bpbtbl6:        dc.b    0,0,0,0,0,0,0,0,0,0,0,0
102:    bpbtbl7:        dc.b    0,0,0,0,0,0,0,0,0,0,0,0
103:    bpbtbl8:        dc.b    0,0,0,0,0,0,0,0,0,0,0,0
104:    bpbtbl9:        dc.b    0,0,0,0,0,0,0,0,0,0,0,0
105:    bpbtbl10:       dc.b    0,0,0,0,0,0,0,0,0,0,0,0
106:
107:    _inittbl:
108:            dc.l    bpbtbl1
109:            dc.l    bpbtbl2
110:            dc.l    bpbtbl3
111:            dc.l    bpbtbl4
112:            dc.l    bpbtbl5
113:            dc.l    bpbtbl6
114:            dc.l    bpbtbl7
115:            dc.l    bpbtbl8
116:            dc.l    bpbtbl9
117:            dc.l    bpbtbl10
```

BPB情報の調査

次に従機側の試作を行います。5月号のプログラム「R.X」をベースに作ります。

前回登場したファイルは次の3本です。このうち，cx.batは主機側に同名のファイルがありましたが，別のものです。筆者はこの名前を毎回コンパイルリンク用のバッチファイルに使用しているためにこのようになっています。詳しいファイル内容については，5月号を参照してください。

r.c　従機側仮想デバイスドライバプログラム
d3.c　通信の入出力ルーチン(5月号のリストを参照)
cx.bat コンパイルリンク用バッチ

変更点ですが，手始めにリスト3の3行を変更してください。これらはともに_drvという変数を使用しています。これを，req－>unitcd＋_drvというかたちに変更してください。

この_drvという変数は従来，SRAMディスクのドライブ番号を格納するために使っていたのですが，今回から複数のドライブを参照するということで，このように変更します。

次に大きな変更として関数の入れ換えがひとつあります(リスト4)。これは先ほどの主機側の変更に呼応するものです。

リスト4の修正が終わったならば，cx.batファイルを実行しコンパイルリンクを行ってください。

従来のr_dskini() 関数はただのダミー関数で正常の帰り値を設定しているだけでした。

変更後は主機のdskini() 関数と呼応してディスク装置を調べBPB情報を主機へ送ります。

BPB情報はDOSコールのGETDPB() を呼んで取得しています。GETDPB() はDPBテーブルというHuman68kの管理するディスク管理情報を取得します。DPBテーブルというのは，本来Human68kが起動したときにBPBテーブルをデバイスドライバより取得し，この情報をもとに作成する情報です。ですから，ここではDPB情報よりBPB情報を逆算して作成してあげることになります。このあたりの処理は468行目から476行で行っています。まあ，機械的な変換といえるでしょう。

表1にBPBの内容を，表2にDPBテーブルの内容一覧挙げておきます。

リスト2

```
--------------------- 変更前 -------------------------
452:/***************************************************************
453:  dskini  ディスク初期化ルーチン
454:***************************************************************/
455:int dskini( req )
456:struct REQ_INI *req;
457:{
458:  req->errlow = 0;     /* 下位バイトエラーコード格納*/
459:  req->errhigh = 0;    /* 上位バイトエラーコード格納*/
460:
461:  req->mxunit = 1;     /* 最大unit  セット*/
462:  req->bpbpoi = &inittbl;     /* BPBテーブルアドレスセット*/
463:  req->devend = &inittblend; /* デバイステーブルエンドアドレスセット*/
464:
465:  return( 0 );
466:}

--------------------- 変更後 -------------------------
452:****************************************************************
453:  dskini  ディスク初期化ルーチン
454:***************************************************************/
455:int dskini( req )
456:struct REQ_INI *req;
457:{
458:  int  sts;
459:  int  n;
460:  char  u_no;
461:  char  drv_flg;
462:  char  **wk;
463:  char  *bpb;
464:
465:  _rs_buf_clr();          /* rs232c buf clr*/
466:
467:  n = (int)(req->reqlen);
468:  req->errlow = 0x03;     /* 下位バイトエラーコード格納*/
469:  req->errhigh = 0x50;    /* 上位バイトエラーコード格納*/
470:
471:  u_no = 0;
472:  wk = (char**)(&inittbl);
473:  bpb = *wk;
474:
475:  OUT232C( 'S' );        /* 送信開始コード送出*/
476:  if( (sts=blk_outl( &n, sizeof(n) ))) {/* リクエストヘッダ長送信*/
477:    ;
478:  }
479:  else if( (sts=blk_out( req, req->reqlen ))) {/* リクエストヘッダ送信*/
480:    ;
481:  }
482:  else {
483:    PRINT( (unsigned char*)"¥n¥r仮想ドライブシステム登録開始¥n¥r" );
484:
485:    while( 1 ) {
486:      if( (sts=blk_in( &drv_flg, 1 ))) {/* bps status flg受信*/
487:        break;
488:      }
489:      else if( drv_flg == -1 ) {  /* normal end*/
490:        if( u_no==0 ) {
491:          break;
492:        }
493:
494:        req->errlow = 0;     /* 下位バイトエラーコード格納*/
495:        req->errhigh = 0;    /* 上位バイトエラーコード格納*/
496:
497:        req->mxunit = u_no;         /* 最大unit  セット  */
498:        req->bpbpoi = &inittbl;    /* BPBテーブルアドレスセット*/
499:        req->devend = &inittblend;/* デバイステーブルエンドアドレスセット*/
40:        break;
501:      }
502:      else {
503:        if( (sts=blk_in( bpb, 12 ))) {/* bpb table受信  */
504:          break;
505:        }
506:
507:        bpb += (char*)(12);
508:        u_no ++;
509:        PRINT( (unsigned char*)("  仮想ドライブを1基追加 ¥n¥r") );
510:      }
511:    }
512:  }
513:
514:  return( 0 );
515:}
```

リスト3

```
--------------------- 変更前-------------------------
327:  DISKRED( (unsigned char*)_rw_buf, _drv, rec, req->dmalen );
381:  DISKWRT( (unsigned char*)_rw_buf, _drv, rec, req->dmalen );
413:  mode = DRVCTRL( mode, _drv );

--------------------- 変更後 -------------------------
327:  DISKRED( (unsigned char*)_rw_buf, req->unitcd+_drv, rec, req->dmalen );
381:  DISKWRT( (unsigned char*)_rw_buf, req->unitcd+_drv, rec, req->dmalen );
413:  mode = DRVCTRL( mode, req->unitcd+_drv );
```

使用方法

●主機側の設定

まず，d0.xを主機のCONFIG.SYSへ組み込みます。

d0.xをシステムディスクのサブディレクトリSYSへコピーしてあることを確認してから，CONFIG.SYSへ次の行を設定してください。

device=¥sys¥d0.x

RS-232Cの設定は仮想デバイスドライバの中で行っているので不要です。もっとも，RS-232C用のデバイスドライバを組み込むのであれば，本プログラム中で行っている設定が上書き変更される可能性があります。このようなときは，SWITCH.Xコマンドにより通信パラメータを本プログラムと同じに設定しておくとよいでしょう。主機，従機ともに同じ設定にしなくてはいけないのはいうまでもありません。

ここで特に重要なのはデータ長が8ビットであること，

およびX制御がnoneであることです。これはバイナリデータを扱うためです。

9600bps

data8,stop1,x-none,パリティなし

のように設定しておきますと，特に指定しない限りX68000起動時に自動的に設定が行われます。

プログラムの動かし方

これで，いよいよ仮想ドライブを動かす準備ができました。まず，従機の電源を入れてください。

次に，今回試作したプログラム「R.X」を実行します。パラメータには仮想ドライブとして使用したいドライブ名を指定します。

例 A>R -DC

ここで気をつけてほしいのは共有を開始したいドライブ名を指定するということです。たとえば，A:，B:，C:という3つのドライブがあったとします。A:はハードディスク，B:とC:はフロッピーディスクです。このときに，B：ドライブから仮想化したならば，

A>R -DB

として起動します。これにより，B:および続くC:ドライブが仮想ディスクとして主機に登録，使用できるようになります。

なぜこのような仕様にしたかというと，通常はハードディスクのドライブ名はフロッピーより前にくるためです。ハードディスクを仮想化するのは危険（クラッシュした場合は）なため，明示的に仮想化を開始したドライブを指定するかたちにしました。

ですから，すべてを仮想ドライブとして登録したい場合は，

A>R -DA

としてください。

次に，主機の電源を入れます。これにより，仮想ドライブは主機の後ろ側に追加登録を行います。

ですから，主機の最終ドライブ名が「D:」であれば，仮想ドライブは「E:」から割り振られます。

このプログラムを終了したいときには，エスケープキーを押してください。強制終了します。

また，このときに，もしも従機の電源が入っていない，またはR.Xが起動されていなければ，この仮想ドライバシステムは登録されません。

今後の課題

当連載もだいぶ煮詰まってきたように思います。仮想ドライブというのはユーザーインタフェイスがほとんどないので，ひとたびできてしまうと，あまり改良のしようがないものなのです。

今後の予定はまだ，はっきりと決めてはいないのですが，2つのアプローチを考えています。

まずひとつがオーバー9600bpsの高速通信。X68000同士であれば38400bpsないし76800bpsの速度がの実験です。

2つ目が，異機種への接続です。現在，ハードディスクの仕様はまだ詳しく調査していないのですが，フロッピーディスクであれば，MS-DOSマシン（PC-9801かIBM系に焦点を絞る）を従機にしてRS-232Cで接続，これのフロッピーディスクをX68000より仮想ドライブとして利用できるはずです。これは，Human68kのフロッピーディスクの仕様がMS-DOSと互換性があるので，理屈のうえでは問題はないのです。加えて今回作ったR.Xは，MS-DOSと互換性のあるHuman68kの機能を用いているので移植は簡単なはずです。

ハードディスクに関しては，MS-DOS自身がバージョンアップを繰り返しているので，ファイル管理システムが変化している可能性があります。変更がファームウェアレベルであり，DOSのファンクションコールから見たとき互換性が維持されていれば，仮想化できるはずです。

リスト4

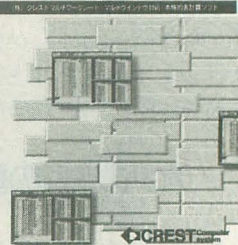
```
--------------------- 変更前 -------------------------
427:/**************************************************************
428:  dskini  ディスク初期化ルーチン
429:**************************************************************/
430:int r_dskini( req )
431:struct REQ_INI *req;
432:{
433:  req->errlow = 0;          /* 下位バイトエラーコード格納  */
434:  req->errhigh = 0;         /* 上位バイトエラーコード格納  */
435:
436:  return( 0 );
437:}


--------------------- 変更後 -------------------------
427:struct  BPB_TBL  {
428:  unsigned short  b_no;         /* セクタあたりのバイト数*/
429:  unsigned char  sct_no;        /* クラスタあたりのセクタ数*/
430:  unsigned char  fat_no;        /* ファット領域の個数*/
431:  unsigned short  rsv_sct_no; /* 予約領域のセクタ数*/
432:  unsigned short  root_ent_no;/* ルートの最大ファイル数*/
433:  unsigned short  sct_max;      /* 全セクター数  */
434:  unsigned char  id;            /* メディアバイト*/
435:  unsigned char  fat_sct_no;  /* 1fatのセクタ数*/
436:};
437:/*************************************************************
438:  dskini  ディスク初期化ルーチン
439:*************************************************************/
440:int r_dskini( req )
441:struct REQ_INI *req;
442:{
443:  int  sts;
444:  struct  DPBPTR  d;
445:  struct  BPB_TBL  bpb_tbl;
446:  int  fat;
447:  int  fat_no;
448:  int  rsv_sct;
449:  char  dsk_flg;
450:  int  drv;
451:  char  d_no;
452:
453:  drv = _drv;
454:
455:  while( 1 ) {
456:    sts = GETDPB( drv, &d );
457:    dsk_flg = 0;
458:
459:    if( sts ) {
460:      dsk_flg = -1;
461:      sts = blk_out( &dsk_flg, sizeof(dsk_flg) );
462:      break;
463:    }
464:    else if( (sts=blk_out( &dsk_flg, sizeof(dsk_flg) ))) {/* bpb tbl送信*/
465:      break;
466:    }
467:    else {
468:      bpb_tbl.b_no = d.byte;            /* セクタあたりのバイト数*/
469:      bpb_tbl.sct_no = d.sec + 1;       /* クラスタあたりのセクタ数*/
470:      bpb_tbl.fat_no = d.fatcount;      /* ファット領域の個数*/
471:
472:      bpb_tbl.rsv_sct_no = d.fatsec; /* 予約領域のセクタ数*/
473:      bpb_tbl.root_ent_no = d.dircount;  /* ルートの最大ファイル数*/
474:      bpb_tbl.sct_max = d.maxfat * (d.sec+1);  /* 全セクター数*/
475:      bpb_tbl.id = d.id;                /* メディアバイト*/
476:      bpb_tbl.fat_sct_no = d.fatlen; /* 1fatのセクタ数*/
477:
478:
479:      if( (sts=blk_out( &bpb_tbl, sizeof( bpb_tbl ) ))) {/* bpb tbl送信*/
480:        break;
481:      }
482:
483:      d_no = 'A' + (char)drv;
484:      d_no --;
485:      printf( "%c: を主機へ仮想ドライブとして登録 ¥n", d_no  );
486:
487:      drv ++;
488:    }
489:  }
490:
491:  return( sts );
492:}
```

新製品紹介

F-Calc for x68k

Yamato Satoshi 大和 哲

以前，Oh!Xでも紹介されたカード型データベース「F-Card」の開発・発売元であるクレストからX68000用の表計算ソフトが発売された。その機能について紹介しよう。

この「F-Calc for x68k」(以下，「F-Calc」)は，現在売られている表計算ソフトとしてはかなりコンパクトなものである。他機種の表計算ソフト，特にWindows用に発売されているソフトでは，フロッピーディスク5～6枚組でハードディスクが必須という表計算ソフトが少なくないのに比べると，フロッピーディスク1枚で，Human68kのシステム，かな漢字変換に使うFEPと辞書，そしてこの「F-Calc」本体とヘルプファイル，それにサンプルデータが入っているといえば，いかに小さいかわかってもらえるだろう。考えてみると，そもそも現在では，システムフロッピー単体で起動してひと通り使えるような構成になっているビジネスソフト自体が，すでに珍しくなっているともいえる。

発売はパソコンショップ，ソフトベンダーTAKERUのほかにPC-VANオンライン通販も行うということである。しかし，プログラム本体とヘルプ，サンプルだけなら容量はさらに小さいので，別の販売方法も可能だろう。例えば，NIFTY-ServeのSOFTEX(オンラインによるソフトウエア販売。ユーザーはモデムでソフトをダウンロードし，代金はNIFTY-Serve利用料金と同じクレジット口座から引き落とされる)などでもよさそうな大きさだ。

もちろんこの「F-Calc」，ハードディスクへのインストールも可能である。ただし，俗にインストーラなどと呼ばれるソフトは入っていないので，自分でハードディスク上にディレクトリを作ってcopyallコマンドなどでコピーすることになる。インストーラがないというのはX68000のソフトではかなり一般的ではあるが，今後はぜひともつけてほしいものだ。

価格は14,800円で，サイズ，値段ともかなりお手ごろ感のあるソフトだといえるだろう。メモリは2Mバイト以上必要。プリンタはシャープCZシリーズ，NECのPC-PR/NMシリーズ，キヤノンBJシリーズ，それからESC/Pシリーズに対応している。

1-2-3互換の操作系

ところで画面写真を見て，PC-98シリーズやFMRなど他機種のビジネスソフトを使ったことのある人は，この「F-Calc」の画面が，PC-98などでかつて高いシェアを占めていたロータス社の「Lotus1-2-3」(以下，「1-2-3」)に似ていると感じたのではないだろうか。そのとおり。「F-Calc」の画面は，FEPの日本語入力モード以外は「1-2-3」の画面とそっくりだ。そして，実はこの「F-Calc」は画面だけではなく，いろいろな面で「1-2-3」をかなり意識して作られているのである。

まずは，操作系。

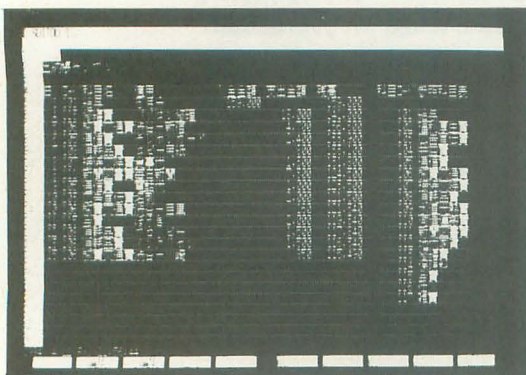
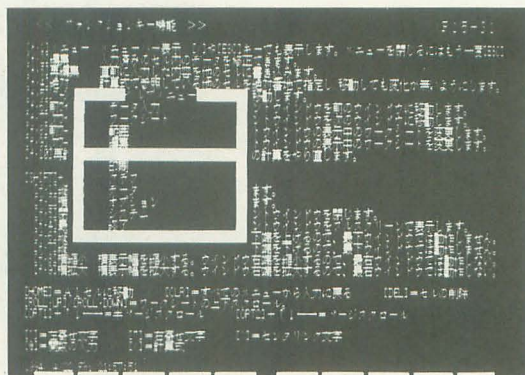
キー操作は，ほぼ「1-2-3」と同じだ。「1-2-3」ではコマンドは「／」(スラッシュ)キーとアルファベット1文字の組み合わせで起動できるのだが，このキーの使い方はもちろん，アルファベット文字の機能へのアサインやファンクションキーの内容も，ほとんど同じ機能になっている。

また，「1-2-3」では「／」キー操作＝メニュー操作となっているが，「F-Calc」のメニューの配置も「1-2-3」とほぼ同じである。

また，「HELP」キーでヘルプ画面を呼び出せ，また，その画面上で，ある機能に対応するキー操作を行うとそれに対応するヘルプが表示される点も同じである。

それから機能面。

「F-Calc」は「1-2-3」とデータ互換も実現している。「1-2-3」で作成したデータファイルを直接読み込んで修正し，再び「1-2-3」形式のファイルとして出力することも可能だ。そのためにワークシート上で使うことのできる最大セルサイズも256列×8192行で同じ。関数も最新の「Lotus1-2-3リリース2.4J」から新たに追加された「@PUREVARS( )」や「@SIGN( )」などはないが，それまでの2.3Jまでにあった関数がすべて同じ機能で存在する。

それから，グラフ作成機能がある。「1-2-3」はもともと，スプレッドシートとデータベース，それにグラフ作成ソフトをひとつにしたい，という要求から生まれたソフトで，スプレッドシートに書かれた数値からワンタッチでグラフを作ることができるという機能がアメリカでの発売当初から

表計算画面も「Lotus1-2-3」そっくりな「F-Calc」

ヘルプ画面から機能別へジャンプできる

好評を博したものである。さらに，それでIBM PCがビジネスパソコンのトップになったというわくつきの機能なのだ。この「F-Calc」にも，同じようにグラフ作成機能が搭載されている。グラフメニューを立ちあげて範囲を設定すれば，棒，円，折れ線，累積棒グラフの表示や印刷をすることができる。

ところで，残念ながら「1-2-3」にあって「F-Calc」にはない機能もある。アドイン機能がそうだ。「1-2-3」では標準の機能のほかに別モジュールで機能を拡張することができるようになっていて，これをアドイン機能と呼んでいる。このアドインモジュールだけでさまざまな製品が市販もされているという，便利でポピュラーなものだ。特にリリース2.4Jでは，キーボードの動きを記録し必要なときに再生できるマクロマネージャが標準でついているが，これが非常に使いやすいものなのである。

もちろん，「F-calc」にアドイン機能をつけても，これらのDOS系のアドインソフトがそのまま動くわけではないし，そもそもこの機能はメモリ上の制約でMS-DOS上ではすべての機能を載せられないための苦肉の策である，という意味もあるかもしれない。が，せめてキーボードマクロは便利なので追加してほしいと思うのだが。

プログラムソースも公開

この「F-Calc」は，関連性をもつデータの処理にも配慮されており，さらにユーザー各自が機能を拡張することもできる。

表計算ソフトで数字や文字を入れるワクひとつひとつをセルといい，セルが集まった縦横の表をワークシートというが，「F-calc」ではこのワークシートを256個まで同時に使うことができる。これはマルチワークシート型と呼ばれるものだ。

また，この複数のワークシートを互いに関連づけて表計算を行う「串刺し演算」もできるので，たとえば，クラス全員の月毎の科目別成績，などといった3次元の配列が必要になる表計算も簡単だ。

また，「F-Calc」はマルチウィンドウも実現している。

マルチワークシート機能だけでは，データ入力は非常に繁雑なこともある。たとえば，ワークシートの表示切り替えができたとしても，データを入力時に画面の切り替えを繰り返していては，データをひとつ飛ばして入力した，などのミスも出てきそうだ。

こんなときは，マルチウィンドウ機能を活用して画面を分割して，複数のワークシートを同時に表示すればそんな間違いもなくなるだろう。同時に開くことのできるウィンドウの数は最大32個である。それでも足りないという人は，実はこのソフトは同梱の申し込みはがきを使えばプログラムソースを入手することができるので，改造してみるのもいいだろう(ちなみにソースの価格は5,000円である)。

また，今回のサンプル版ではまだ試すことができなかったが，セルを移動するたびにその数値などを読み上げてくれる音声読み上げモードなどもつくはずである。私にはあまりメリットがあるようには思えないが，使い方次第では便利な機能だろう。ほかにもX68000ではPC-98などに比べて画面が広い分，一度に多くのセルを見られるというメリットもある。PC-98だと7列×20行しか見られないが，X68000の「F-Calc」なら9列×26行表示される。

最後に

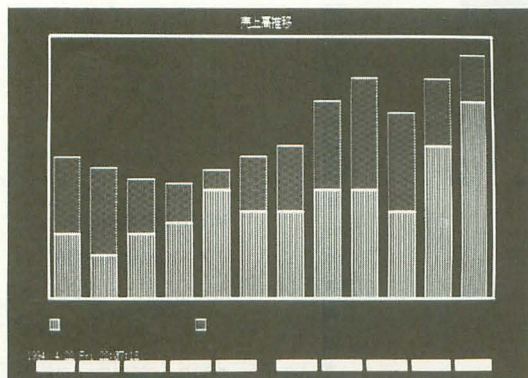
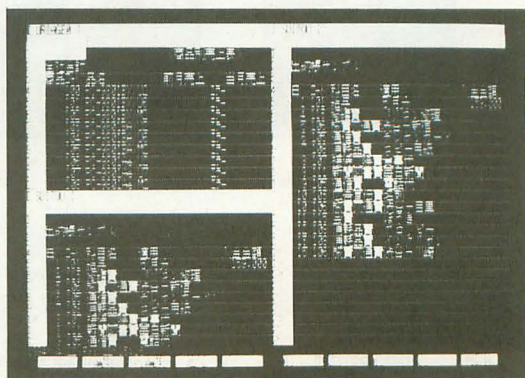
この「F-Calc」，小さい割にはよくできていると思われる。何をするにも軽くて，10MHzのX68000でもまったくストレスなく使える。キー操作も「1-2-3」を参考にしているだけあっておそらく会社の事務嬢(それもお局様クラス)なら目にもとまらぬ速さで表を作ってくれることだろう。

いまさら，「1-2-3」コンパチ？という声も聞こえてきそうだが，会社の事務部門では，まだまだ古い286マシンで「Lotus1-2-3」などを使っていることが多いようだ。PC-98などでも，現在は売れるマシンのほとんどが486マシンで，ソフトも個人ベースではグラフィカルなインタフェイスを活かしたマイクロソフトの「Excel」や，これまでの表計算ソフトの常識であるセル番地という概念をなくしてしまったロータスの「Improv」などの新しいものが普及しているにもかかわらず，である。

もちろん，それは設備として買ったパソコンが税金の関係で何年もリプレースできないため，ソフトも，古いマシンでも実用になる古いものを使わざるを得ないとか，ソフトを変えるためには使う人の教育も同時に行わなければならず，結構な費用がかかるなどの事情があるからなのだが，そういう会社に勤めていて家のX68000でも使いたい，という場合には「F-Calc」はうってつけのパッケージソフトだろう。

ただ，キャラクタインタフェイスの画面かつキーボード操作のインタフェイスにしては，目的の機能を使うまでのストロークは少ないが，「1-2-3」や「F-Calc」を使ったことのない人にとっては，やはりやや難しいのも事実だ。また，マニュアルもB5判100ページあまりの本1冊で，かなり簡単に書かれている。関数の使用例や索引もないので，何も知らない人が使いこなすには市販されている「1-2-3」の入門書を買うなどの必要が出てくるだろう。だから，初めて表計算ソフトを使う，などという人にはあまり向かないかもしれない。

次は，SX-WINDOW上でグラフィカルなユーザーインタフェイスを使った表計算ソフトが欲しいものである(私は「Excel」ユーザーなので，個人的には「Excel」ライクなものがいいな)。

X68000用　3.5/5"2HD版	14,800円(税別)
「F-Card」ユーザー特価	9,600円(税・送料込)
ブラザー工業(TAKERU)	14,800円(税込)
クレスト	☎03(3418)5993
ブラザー工業(TAKERU)	☎052(824)2493

こんなグラフも範囲設定ワンタッチで描ける

画面を分割表示できるマルチウィンドウ機能

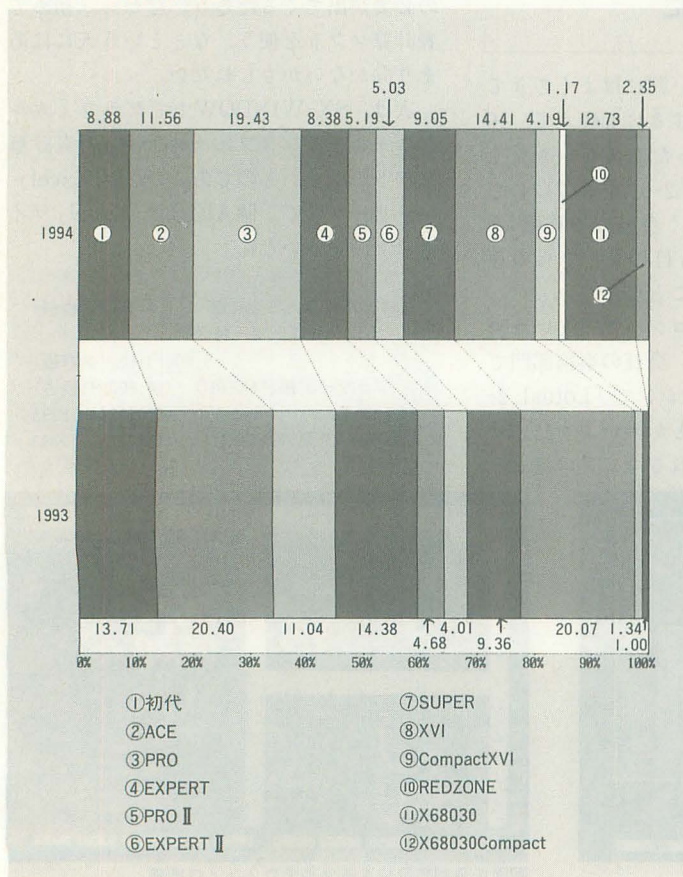
第5回Oh!Xアンケート分析大会

Takahashi Tetsushi 高橋 哲史

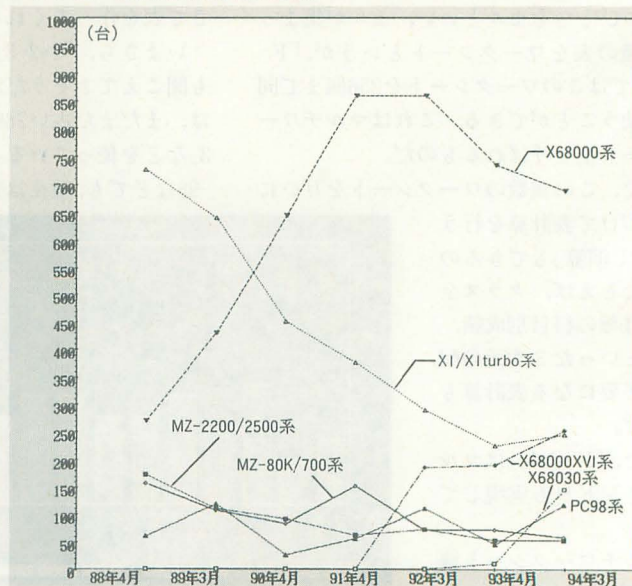
毎年恒例アンケート分析。Oh!Xの読者はどんなことを考えているのでしょうか。
今年は，荻窪圭氏から高橋哲史氏にバトンタッチして行われました。さて……。

桜の花も散り，街は初夏の彩りをたたえる今日この頃。読者の皆様はいかがお過ごしでしょうか？ 今年もOh!Xアンケート分析大会の季節がめぐってまいりました。前年まで分析大会の舞台であった「大人のためのX68000」連載終了にともない，荻窪師匠に代わりましててこの私高橋“いつまでプーなの？”哲史が集計から分析まで担当させていただきますので，しばらくの間のおつきあいのほど，どうぞよろしく。

まずは機種別所有台数

さて今回は，1994年3月号のアンケートのなかから500枚を無作為抽出して集計を行っております。ちなみに例年より200枚多いです。いつもより余計に数えておりまぁ〜すといった感じですかね。

それではさっそく，アンケートの項目順にデータを見ていきましょう。まずは表1，読者に占めるX68000ユーザーの割合です。500人中481人，実に96.2%となりました。一昨年，昨年はそれぞれ81%，93%でしたから，順調に増え続けています。記事内容も最近はすっかりX68000一色になってしまいましたからねぇ。

次はグラフ1。これはX68000の機種別所有台数の割合を表しています。ユーザー481人のなかには複数台を所有しているなんてリッチな人もいますので，全体の台数は597となっています。

ざーっと眺めると，昨年まで多かったACE系EXPERT系がちょっとひっこんで，その分，X68030系が伸びているのがわかります。この辺は買い替えが順調に進行していることを示しているのでしょう。XVI系は台数は着実に増えているのですが，割合としては減少しています。なにせ昨年は約20%と5人に1人がXVIユーザーだったのですから，これはしかたのないことですね。CompactXVIも急速に伸びていますし，REDZONEも新登場ながら健闘しています。やっぱり24MHzの誘惑には耐えがたいものがありますしね。それにしても，昨年は1%だったX68030所有者が今年は一気に約13%！ 68にかけるユーザーの熱い想いが伝わってきますね。POWER TO MAKE YOUR DREAM COME TRUE……こういったユーザーを裏切らないように，これからもサポートお願いしますね，シャープさん。

さて，グラフ2は，1988年からの機種別所有台数の推移を示しています。X1/X1 turboのラインをたどっていくと盛者必衰が実感できますね。X68000系は若干の落ち込み傾向になっていますが，これは10MHz機のことで，XVI系や030系を合わせると全体的な所有台数は増えていることがわかり

グラフ1 X68000機種別ユーザーの割合

表1 X68000の所有者

	人数	割合
68ユーザー	481	96.2%
非68ユーザー	19	3.8%
合計	500	

グラフ2 機種別所有台数の推移

ます。なんだかんだといいつつ，皆さんX68000を手放せないわけですね。

システム環境は向上しているのか?

これから先のいくつかの集計は，481人のX68000ユーザーの回答によるものです。ひとりでたくさんもっている人は，主に使っているマシンについて答えてもらいました。

ではパソコン環境から。環境といえば，まずメモリ容量です(グラフ3)。SX-WINDOWの普及や動画データの増加など，とにかくメモリはあればあるほどよいといった状況ですが，相変わらず2Mバイトユーザーが多いことがわかります。さすがに1Mバイトユーザーはほとんどいません。次いで4，6Mバイト，そしてそのあとがいきなりフル実装の12Mバイトになっているところが X68000らしくていいですねぇ。最近はX68000でもSIMM増設できるボードが発売されましたから，メインメモリはどんどん増えていくことでしょう(しかし私もいまだに2Mバイトだったりして。なんといっても先立つものが……)。

次にハードディスク容量です。少し意外だったのは普及率です(グラフ4)。ハードディスク所有者は380人と，全体の79.5%でした。いまやハードディスクがなかば「必需品」となりつつある今日この頃，80%は軽く超えるくらいの所有率では，と予想していただけに，これはちょっと意外な結果でした。実際の容量がグラフ5。80Mバイトクラスが多いのは内蔵タイプが増えたため。外付けでは相変わらず240Mバイト近辺のクラスが人気を集めています。しかしすでに1G以上のものを13人の方が所有しているところに，本格的な大容量時代の幕開けを感じますね。そのうちみんなギガ単位でも足りないなんていいだすんだろうなあ……(人間の欲望はとどまるところを知りませんね)。ちなみに，私が愛用しているリムーバブルハードディスクの使用者はわずか4人でした。もはや完全にMOに駆逐されてしまった模様です。とほほ。

グラフ4　ハードディスク所有率

続いて周辺機器です。グラフ6をご覧ください。集計していて，SCSIボードと光磁気ディスクを同時に所有している人が多かったのが印象的でした。数値演算プロセッサやスキャナの所有率は比較的高いほうなのかな? やはりダントツはジョイスティックでした。

所有ソフトのベスト10は，表2のとおりです。コンパイラが1位というのはX68000らしい結果ですよね。SX-WINDOWに限り，バージョンごとの集計になっています。

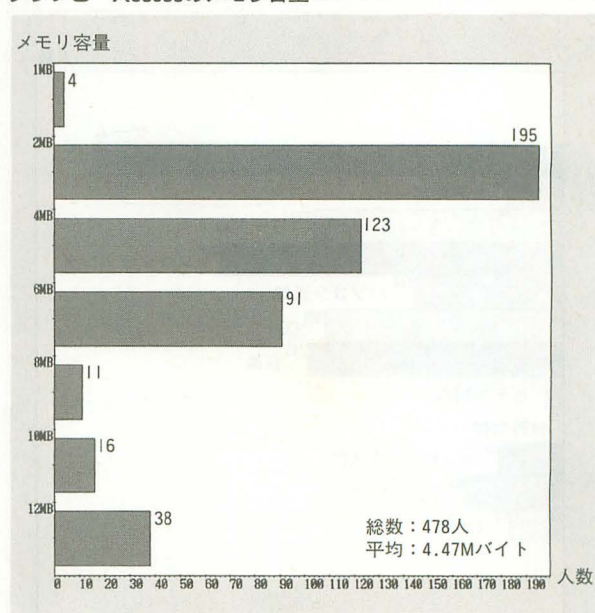
そして，S-OSは，ユーザー数67人で全体の13.4%，とさすがに一時期に比べるとパワー不足の感は否めませんでした。しかしここではX1やMZがいまだに健在なのが注目すべき点でしょう。S-OSの世界には本当にパーソナルコンピューティングの本質があるような気がします(と，まったくプログラムをしなくなった私がいうのもおこがましいですが……)。皆さんも機会があればS

グラフ3　X68000のメモリ容量

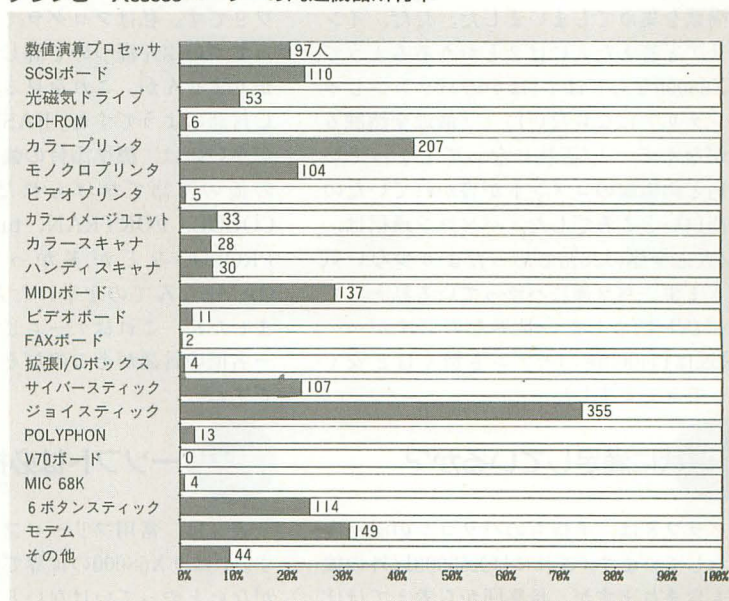
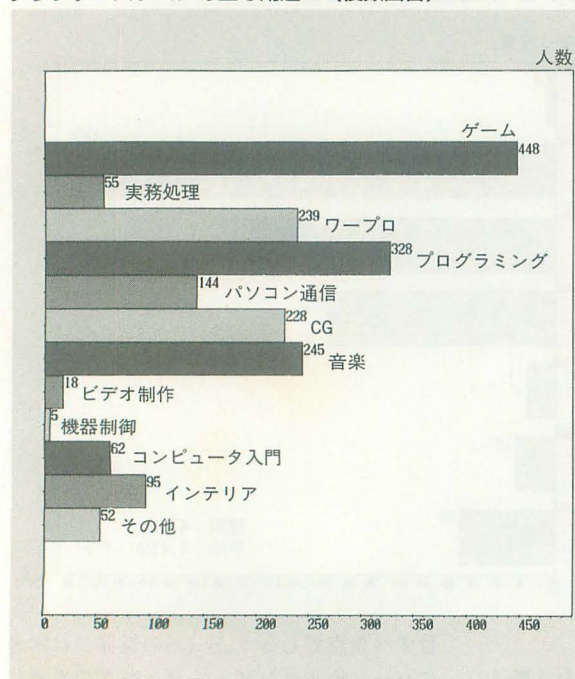
表2　使用ソフトのベスト10

順位	ソフト名	人数
1	C compiler	290
2	Z-MUSIC	280
3	SX-WINDOW ver3.0	247
4	DōGA CGA system	232
5	Z's STAFF PRO-68K	124
6	SX-WINDOW ver2.0	87
7	MUSIC PRO-68K	86
8	MATIER	78
9	NAGDRV	75
10	SX-WINDOW ver1.0	73

グラフ5　主に使用しているX68000のハードディスク容量

グラフ6　X68000ユーザーの周辺機器所有率

グラフ7　パソコンの主な用途　(複数回答)

グラフ8　現在使用しているX68000について　(○△×の3段階評価)

-OSワールドに触れてみることをお勧めします。きっと，目から鱗がぼろぼろ落ちることでしょう。

X68000は何に使われているのか?

続いて用途です(グラフ7)。ゲームがトップなのは例年のことですが，続いてプログラミング，CGワープロ，音楽と続くのがX68000らしいクリエイティブなところです。やはり何かすごいものが生み出せそうな気にさせてくれるところがありますよね♪(CGや音楽は私もハマりまくっています)。昨年は冗談で「インテリア」と書いてくる人も多かったのですが，今年も52人と結構票を集めてしまいました。ただ，インテリアと答えた人には2とおりあるようで，「X68000ならでは!(ほかのパソコンじゃインテリアにならない)」と「最近全然触る暇がなくて……(置物になってしまっている)」と両極端のコメントが書かれていたのが面白いところでした。パソコン通信は，144人と全体の28%で思ったより少ない気がします。パソ通にハマっている私としてはぜひお勧めしたい世界なのですが……(最近は14400bpsのモデムも驚くほど安くなってきてますし)。

現状に満足しているか?

グラフ8は，手持ちのパソコンの満足度を示しています。これにはX68000以外の機種も含まれますが，母集団から考えてほぼX68000に対する評価と考えていいでしょう。アンケートでは○△×の3段階で評価してもらいました。

ゲーム環境やプログラミング環境などはおおむねみなさん満足しているようです。しかし，ビジネス環境，周辺機器の充実度，メーカーサポートには不満を感じているユーザーも多いようです。確かにビジネスにX68000を使うのはかなり厳しいのが現状でしょう。シャープさんにもなんとかしてもらいたいところです。しかし使いやすさとデザインに関しては圧倒的に不満が少ないところが，X68000らしいといえるかもしれません。

プログラミング言語は?

使用プログラミング言語についてがグラフ9です。私はプログラミングをしなくなってずいぶん経つので詳しい事情はよくわかりませんが，これを見ると時代はやはりC言語のようですね。BASICを使用する人が多いのは，標準添付の強みでしょう。その他の言語で挙げられたものとしてはCOBOL，FORTRAN，turboPASCAL，PROLOGなどが多かったようです。GAMEなんてのを挙げた人も3人ほどいましたが，これはゲームじゃなくてそーゆー名前の言語があるのだそうです。面白いですね。

フリーソフトは必携品

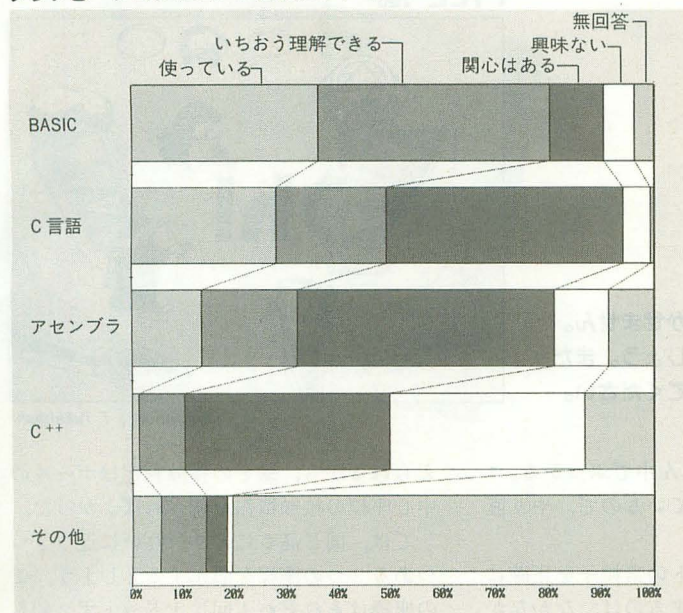
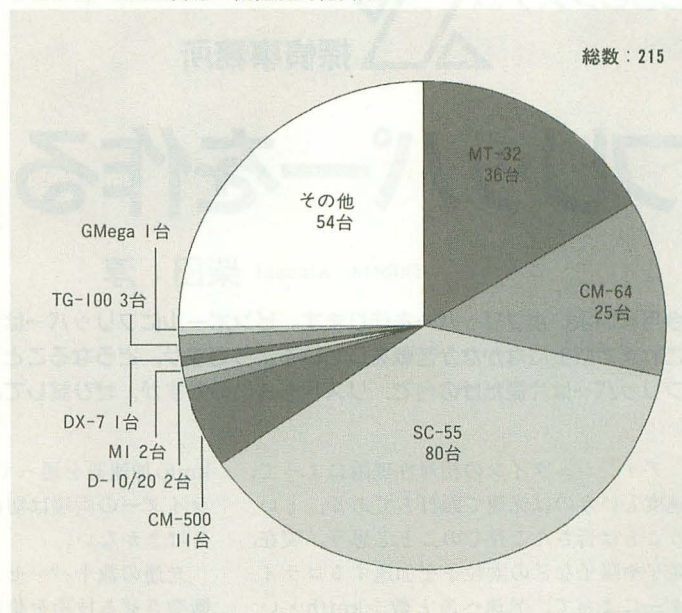
表3は，常用フリーソフトのベスト10です。やはりX68000の世界ではフリーソフトがないとやっていけないというのが現状でしょう。下手すると市販のものより質がよかったりしますからね(この辺にX68000ユーザー気質が出てるなあ……)。

パソコン通信をやっているかとか，どんなX68000友達がいるのかによって，構築される環境がまったく違ってくるのがフリーソフトの面白いところです。私も，たまにしか遊びに行かない友達のところでは「これが同じX68000か!?」と，環境の違いに愕然とすることも多々ありますし。今回いちばん意外だったのは，ディスクシェルとしてTFがDIを圧倒していたことです。自分の周りの環境を見ていて，だいたいDI派とTF派とは半々くらいかな，と思っていたのですが……。いやいやわからないものですね。ちなみに私はDI派です。ま，このあたりは各人が使いやすいものを使っていけばいいわけですし……(よく自分の環境を人にゴリ押しする方がいますが，あれは感心できませんね。世界はあなたを中心に回ってはいませんよ)。

HASにHLKにDISは，プログラマの三種の神器といったところでしょうか。エディタも好みがはっきり別れていて面白かったです(「その他」の欄にSuperEDを挙げる方のなんと多かったことか)。あと16色グラフ

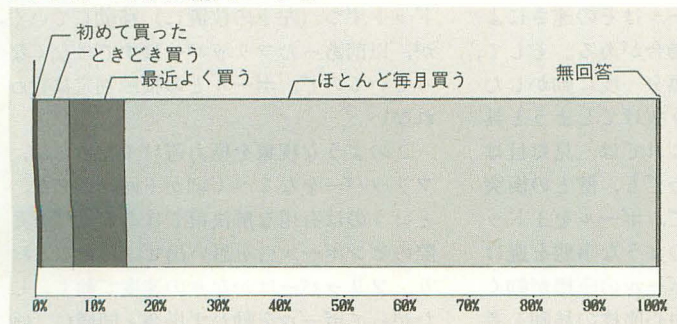
表3　使用しているフリーソフトのベスト10

順位	ソフト名	人数
1	LHA	367
2	PCM8	268
3	APIC	235
4	LZX	218
5	PI	213
6	HAS	213
7	HLK	209
8	GCC	151
9	MAG	148
10	TF	129

グラフ9　プログラミング言語について

グラフ10　MIDI音源　機種別所有台数

ィックフォーマットとしてPIがMAGに勝っているというのもX68000ならではという感じがします。

しかし，「その他」の欄のコメントで「MuTERM, ZED, RCD, STED2, MMDSP, AJAG, 9 SCDRV, TwentyOne, DINIT, GRAD, SXSI……とても書き切れません」と，記入してある方がいました。これこそ，いまのX68000事情がよくわかる一文だと思いませんか。

やはりDTMはRoland強し

さて続いては，MIDI音源の勢力図です。グラフ10をご覧ください。グラフ6の周辺機器のMIDIボード所有率(27.4%)を見てもわかるとおり，すっかりMIDIは市民権を得たようです(最近はRS-MIDIしてる人も多いはずだから，実際はもっとMIDI人口は多いでしょう)。そして，そのなかでもやっぱり強いのがRoland。所有音源のほぼ7割がRolandユーザーということになっています。今春のSC-88発売でRolandユーザーはまた増加の一途をたどると思われますが……。GSだGMだのと混迷するDTM業界はいったいどこへいってしまうのでしょうか？　ま，マイナー機種TG-100ユーザーの私にはカンケーないことですが。

その他の音源としては，X3やE-mu，ASRシリーズなんかがありました。あ，あとAKAIのSシリーズを所有していれる方も2～3人いらっしゃいましたね(サンプラーは私もすごく欲しいです……)。

貴方にとってOh!Xとは？

さて最後に，本誌Oh!Xに関する2項目をまとめてお送りします。グラフ11をご覧ください。ほとんど毎月本誌を買ってくださっている読者が80%という素晴らしい結果になっています。んー，予想通り(？)。アンケートを見ていると「ここ3年くらい毎月買ってます」というような，Oh!Xからの読者が7割くらいで，残りがOh!MZ時代からの読者という感じです。そういう方々は必ず「MZから読んでます！」とアンケートでアピールしています(気持ちはよくわかります)。私も1984年から読んでいたので，もう10年になってしまうんですね……(しみじみ)。

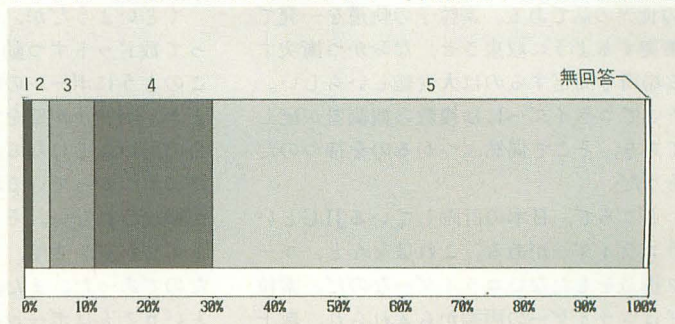
そして最後に，Oh!Xの必要度を表わしているのがグラフ12です。1～5の5段階評価をしてもらいました。必要度4，5を挙げてくださった方が全体の87%。スタッフとしては非常に嬉しいですね。ただ，必要だと答えてくださった方のなかにも「X68000関連の情報が入るのはOh!Xと電脳倶楽部だけなので」といった消極的な必要性を感じている方も結構いらっしゃいました。また不必要だという方には「最近X68000の記事ばかりなので」という生粋のMZマニアの方も多々見受けられました。やはり時代の流れなんでしょうけど……。難しいものですねぇ。

ということで，第5回アンケート分析大会はこれにて終了です。はっきりいって1人で500枚すべての集計は死ぬかと思いました。5月号の「ちゃだワ」に続き，まさに読者パワーに圧倒された2カ月間といっても過言ではなかったでしょう。いやー，読者様様です。それではまた来年お会いいたしましょう。さよーならー。

グラフ11　Oh!Xの購入について

グラフ12　Oh!Xは必要なものですか？（不必要を1とした5段階評価）

こちらシステム X 探偵事務所

FILE-XIII

フリッパーを作る

Shibata Atsushi 柴田 淳

今月はいよいよフリッパーを作ります。ピンボールにフリッパーは欠かせません。これまで以上になかなか苦戦をしているようですが，どうなることでしょう。まだフリッパーは片側だけの台で，リストも長いのですが，ぜひ試してみてください。

illustration : T.Takahashi

アインシュタインの相対性理論によって，速度というのは光速で頭打ちである，ということは皆さんご存じのことと思う。現在，電子や陽子などの素粒子を加速するコライダーによって，光速へあと数十km/hというところまで電子を加速できるのだそうだ。このくらいまで光速に迫れるのなら，残りの速度はコライダーのループ自体を動かしてしまったほうが速いような気がするが，実際はそうそううまくはいかない。というのは光速に近づけば近づくほど，加速にかかるエネルギーは膨大なものになるからだ。ちなみに日本の雄“トリスタン”では，実験のため素粒子を加速するのに，1カ月も大きな電力をかけ続けることがあるという。

なぜこんなにまで粒子を加速しなければならないかというと，未知の粒子を見つけるためである。物質のもつエネルギーは質量と速度に比例する。そしてエネルギーは物質に転化し得る。そこで加速された素粒子をぶつけ，新物質を作ろうとするのである。比較的低いエネルギーで生み出される素粒子はあらかた見つけられており，残る素粒子を見つけるためには，当然より大きなエネルギーが必要なわけだ。

たいていのコライダーは円状である。超伝導磁石を装備したループを素粒子が駆け巡り，あるときは静止した物体に，またあるときは反対方向に加速された素粒子にぶつける。ただぶつける，といってもミクロの世界の話である。素粒子の軌道を一発で衝突するように収束させ，なおかつ衝突する場所を固定するのは大変難しいらしい。そこでコライダーには複数の観測器が配してあり，そこで偶然ぶつかるのを待つのだそうだ。

ところで，日本の計画しているJLCというコライダーがある。これはなんと，ループ構造をもたないコライダーなのだ。素粒子はコライダーの両端から入れられ，数十kmの加速器を通って真ん中でぶつかる。コライダーの両端は切れているので，やり直しはきかない。

光速の数十パーセントの素粒子を正確に衝突させる技術を想像するのは，なかなかスリリングである。僕がいまやっている，ピンボールのシミュレーションより格段に難しいことは確かだ(当たり前だが)。日本人ってなかなかすごいぞ。

動対動の衝突判定

そろそろ本題に入ろう。今回はフリッパーを作っていく。そのフリッパーのために必要な処理のうち，ボールとフリッパーの反射と，ボールが弾かれる，という2つの処理は，前回にバンパーを作ったときに行った処理を応用（というよりほぼそのまま流用）すればいい。

いままでは壁やバンパーのように，静止したものと動くボールの衝突しか扱ってこなかった。ところが，フリッパーはボタンを押すと軸を中心に回転をする。つまり，動くもの同士の衝突判定を行わなくてはならず，これには新しいノウハウが必要である。

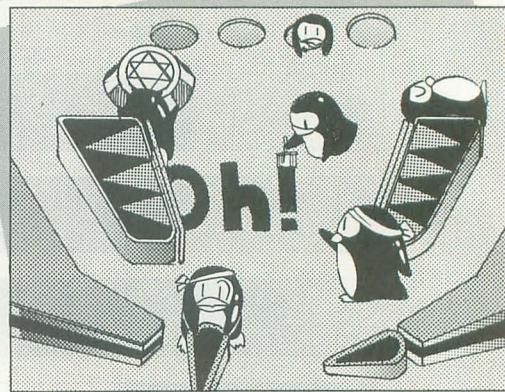
ところで，動くもの同士の衝突判定とは，いったいどういったものだろうか。問題点を明らかにするために，壁との衝突判定を思い出してみよう。

くどいようだが，ボールはその速さによって数ドットずつ動く場合がある。そして，このようにボールの座標を一度に動かしたとき，ボールが壁をすり抜けてしまうと具合の悪いことになる。これでは，見た目は壁にぶつかっている場合でも，壁との衝突が検出されない。そこで，ボールを1ドットずつ移動させて，このような事態を避けたのであった。また，ボールの座標が動くということはボールの中心座標の移動と考えられるから，壁との接触判定はボールの中心座標の接触情報を調べればよかった。

では，同じ高さにあり，互いに近づきつつある2つの座標を想定するとしよう。この座標はそれぞれ1回に1ドットずつ動くものとする。で，1ドットずつ動くのであれば，近づく2つの点の座標が重なったことを100％検出できる，と思われるかもしれないが，実はそうはならない。

試しに，数直線を頭に思い浮かべてみよう。1つ目の座標は数直線の－4の位置にあり，2つ目の点は5の位置にあるとする。このような2つの点が，原点方向に向かい1ドットずつ進んでいくとどうなるだろうか。2つの点がいちばん近づくのはそれぞれ0，1にあるときである。で，ここから左の点は1増え，右の点は1減るのだから，2つの点は1，0に移動する。2つの点が重なったかどうかで衝突を判定する方法だと，このように見た目は衝突していても点がまんまとすり抜けてしまう場合を判別できない。

ボールとフリッパーの衝突判定を行おうとすると，このような「すり抜け」が起こる。図1にその現象を模してみた。図中の右下を向くフリッパーと右上のボールは，ボールがすり抜ける直前のフレームの位置を示す。次のフレームでフリッパーは上(真横の位置）に上がり，ボールは移動ベクトルから計算した1フレーム分の移動量を1ドットずつ（左下の位置に）移動していく。が，以前あったフリッパーはすでになくなっているので，ボールとの接触判定は行われない。

このような現象を極力避けるためには，フリッパーをなるべく細かく回転させる，というのは有用な解決策ではある。だが実際のピンボール台を思い出せばわかるとおり，フリッパーはかなりの速度で動く。したがってボールを動かす場合と同様に，画

面上からは見えないような，内部的な処理の導入を余儀なくされる。しかしボールのように1つのドット（中心座標）についてだけの処理ならまだしも，フリッパーくらいの大物を内部的に細かく動かすのは骨が折れそうだ。

ここで少し頭を使う。ピンボール台の壁などの固定物とボールが接触したかどうかを調べるために設けた接触情報マップというのは，描き出された画面とはまったく別に用意されたものである。バンパーを作ったときは，これを利用して実際にバンパーのある位置にはバンパーの所在を示す情報だけを書き込んでおき，接触情報マップの外側にあたる場所にバンパーの角度を書き出す，という手法をとった。

ではこれと同じ方法論で，フリッパーをボールがすり抜ける現象を解決できないだろうか。画面の表示内容と接触情報は厳密に対応している必要はないのだから，ボールがフリッパーをすり抜けないように接触情報マップを書けばいいのだ。繰り返しになるが，ボールがフリッパーをすり抜けてしまうのは，動くフリッパーの「軌跡」に隙間があることが原因である。では，隙間がないようにフリッパーの接触情報（つまり角度の情報）を書いてしまえばいいのではないだろうか。

壁などの接触情報は，壁からちょうどボールの半径分だけ離れた位置に書かれていた。それに対しフリッパーの接触情報は，画面に表示されているフリッパーよりも広い範囲に書くことにしよう。つまり，フレーム間にフリッパーが移動しているはずの部分（フリッパーの軌跡にあたる）に角度の情報を書き込んでおくのである（図2を参照）。こうすることによって，見かけ上はフリッパーとボールの衝突が起こるべきすべての位置で，2つの物体の衝突を検出できるはずだ。

反射の裏と表

さて，フリッパーとボールの衝突について，もう少し考察を進めてみよう。ピンボールで遊んでいると，ごくたまにフリッパーの裏側にボールが当たることがある。で，この裏側にボールが当たった場合，ボールの動きはフリッパーの動作によってさらに2種類に分かれる。

1つは，フリッパーが振り切って定位置に戻る途中で，ボールがフリッパーの裏側に当たる場合。このときはボールは下向きに弾かれ，ボールを1つ失ってしまう（運がよければ復活することもある）ことになる。これは，通常フリッパーがボールを弾くことの逆であると考えられる。

2つ目は，フリッパーの裏側にボールが当たるとき，フリッパーが正方向に振られている途中の場合，ボールはどのような動きを示すだろうか。まず思い出していただきたいのがバンパーの処理だ。ボールがバンパーに当たると，まずバンパーの面との反射によってボールは力を受ける。そしてバンパーは面と垂直な方向に力を発し，それがボールをさらに押し上げるのであった。

フリッパーの場合，この「ボールを弾く」力となるのはフリッパー自体の回転運動である。しかし，このフリッパーの運動によって働く力は回転方向にしか働かない。すると，フリッパーが正方向に回転しているときにボールが裏側に当たると，ボールは反射による力しか受けないことになる。ここに，壁・バンパー対ボールの衝突を扱ってきたときには問題にされなかった，新たな「場合分け」の必要性が出てくる。

ボールがフリッパーに当たった面によって処理を分けるためには，フリッパーとボールの衝突が検出されたときにフリッパーのどちらの面に当たったのかを判断できればいい。まず思いつくのが，ボールの当たった場所の角度によって表裏を判定する方法だろう。フリッパーの面の角度は接触情報マップにあらかじめ書き出してあると仮定すると，ボールの中心座標にあたる部分を参照し，得られた角度からフリッパーの表裏を判定するわけだ。

だが，よく考えてみるとこの方法には難点があることに気づく。というのは，接触情報マップの角度は，実は半分の角度，つまり0度から180度までしか用意していないのである。

たとえば，真横に延びた壁があるとする。この壁の角度は180度ともいえるし，0度ともいえる。この壁にボールが反射するとき，反射後の方向を決めるのは，壁を0度ととるか180度ととるかではない。反射後のベクトルを求めるために，壁の角度を2倍して三角関数を得たことを思い出すと，180度のほうは2倍されて360度となり，結局0度と同じになることから，このことが理解できる。角度の表現方法はどちらでもいいのだから，混乱を避けるため重複する部分は使わず，そのぶん角度を細かく分けたほうがいいわけである。

図1　ボールがフリッパーをすり抜けてしまう場合

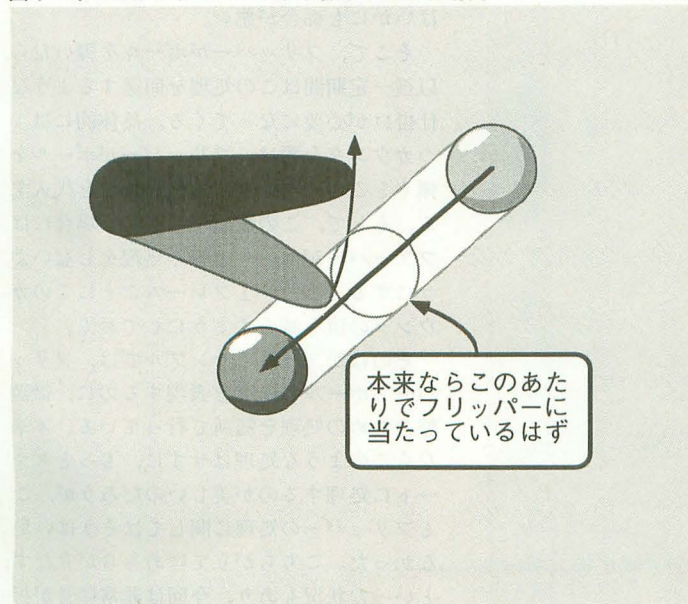
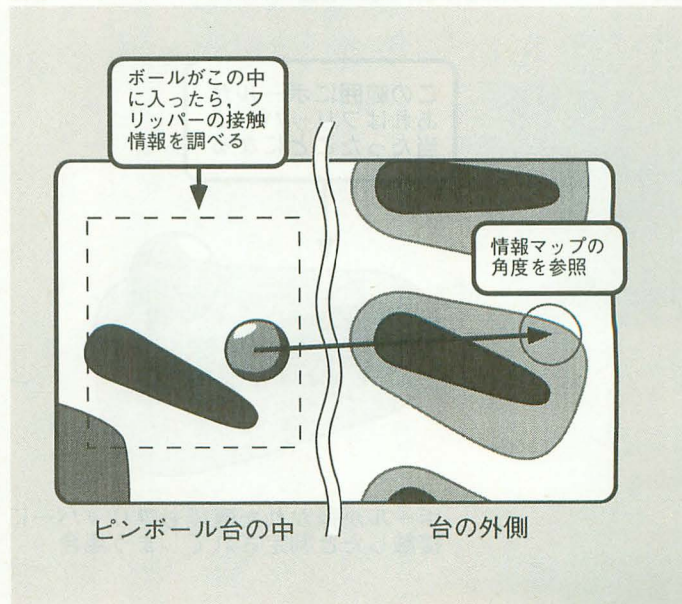



図2　フリッパーの接触情報マップの例

要するに，この場合の壁の角度は（フリッパーの角度も同様に）2つの意味をもつわけだ。このように裏表がいっしょくたになっていては，角度からフリッパーの面の向きを決定することはできない。

そこで，ない情報は新たに作ることにする。フリッパーの角度の情報とは別に，面の向きを表す情報を接触情報マップに書き込むのだ。フリッパーにボールが接触したことがわかった時点で面の向きの情報も読み，そのあとボールを弾くか，あるいは単なる反射の処理をするかを決めればいい。

このように，壁の角度といった単純な情報だけでなく，さまざまな付加的情報を重ねることができるのが，裏マップをもつことの強みである。裏マップについての定義というのはただ1つ，1ドットが1バイトに対応している，ということだけなのだ。そこに書き込まれた情報を解釈するのはプログラムであり，したがって工夫次第でいろいろなことができる。またこうしておけば裏マップの仕様も比較的楽であり，プログラミングの生産性も上がるという寸法である。

フリッパーの実際

次に，フリッパーの処理をもう少し具体的に見てみよう。とはいえあまり細部に触れているときりがないから，ポイントになる部分だけ。

まず，今回のサンプルでは，画面上のフリッパーのある位置の裏マップには，情報はなにも書き込まれていない。これはフリッパーは動くものであり，固定した壁のように角度の情報が一意に決まらないからである。

実際のフリッパーの角度情報は画面の外にあり，ボールがフリッパーに近づくと，ボールの中心座標にオフセット値を足して角度情報を参照している。このとき，同時にフリッパーの面の向きの情報を得ているのはすでに説明したとおりだ。

サンプルでは，フリッパーは5種類の角度をとる。だが，実際に情報として書き出してあるのは，3種類だけである。いちばん下向きのフリッパーから，真横を向いたフリッパーまでで，残りはこれを上下に反転して使用している。ちなみにこれを左右に反転すれば，もう一方のフリッパーの情報として流用できるのはいうまでもない。

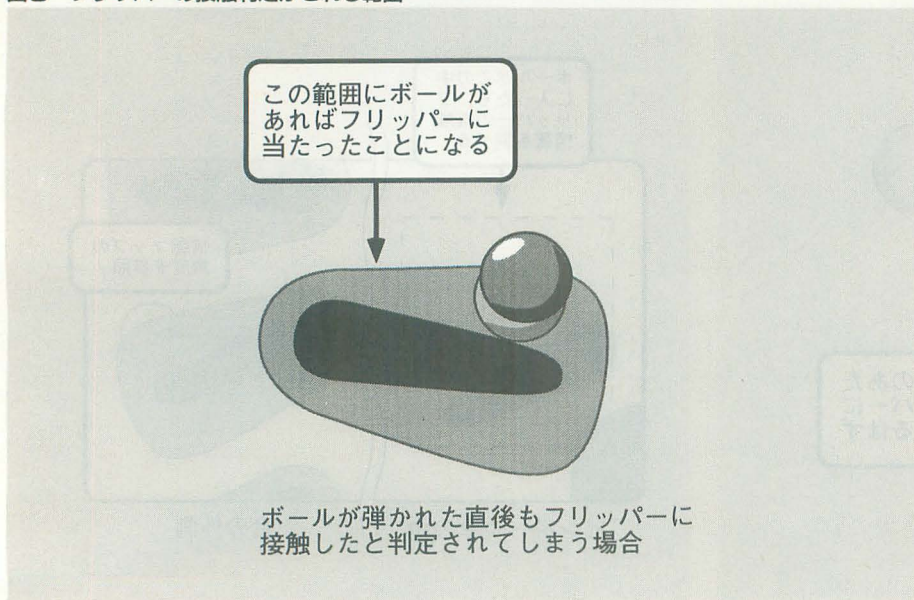
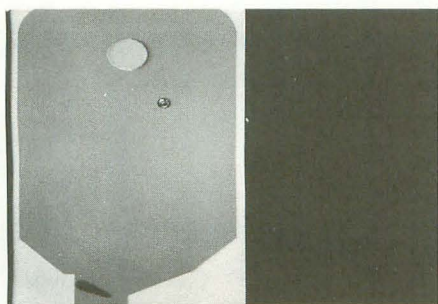
また，実際にフリッパーのある場所に角度の情報を設けないことで，処理速度も稼げる。フリッパーの接触情報を画面の外に一括して書き出しておけば，フリッパーの角度によって変えなければならないのは接触情報を引き出すためにボールの中心座標に足すオフセットの値だけだ。これと似たような方法論で，画面描き換えにスプライトなどを使えば，いくらでも大きな物体をピンボール台で動かすことができる。

ボールとフリッパーが接触していることが検出されると，フリッパーの角度，接触した面の向きなどの情報を読み出す。この情報を元に，ボールの動きを計算するわけだ。

ここで，ボールがフリッパーに弾かれたあとのことを考えてみよう。まず，フリッパーの接触情報は，ボールのすり抜けを避けるために実際のフリッパーよりは広くとられている。そして，ボールは壁などのピンボール台にあるものとの接触を100%検出するため，1ドットずつ動いているのであった。すると場合によっては，フリッパに弾かれた直後にもまた接触が検出されてしまう場合がある（図3を参照）。これではいかにも都合が悪い。

そこで，フリッパーがボールを弾いたら，以後一定期間はこの処理を回避するような仕掛けが必要になってくる。具体的には1つカウンタを設け，フリッパーがボールを弾くとこのカウンタに適当な数値を代入する。そして，この数値が0でない場合にはフリッパーがボールを弾く処理をしないようにする。また，1フレームごとにこのカウンタの値を減らすようにしておく。

そのほか，今月のサンプルでは，フリッパーとボールの衝突を表現するのに，微調整のための処理を随所で行っている。本来ならこのような処理はせずに，もっとスマートに処理するのが美しいのだろうが，ことフリッパーの処理に関してはそうはいかなかった。こちらが立てばあちらが立たず，といった状況もあり，今回は非常に骨が折

図3 フリッパーの接触判定がされる範囲

サンプル画面

れた。

今回触れたフリッパーの処理は，ピンボール制作の1つの山場であった。実際，サンプルには，ここで触れなかったたくさんの細かな技術が詰め込まれている。コンピュータのゲームというのは（特にアクション性の高いものほど），さまざまな技術の積み重ねによってできるものなのだ。

サンプルについて

今回のサンプルは，フリッパーの接触情報マップが必要だったこともあり，入力方法がいつもと違う。入力される方は，下の囲み記事を参考にしてほしい。

いつものごとく，実行するとマウスカーソルの位置にボールが現れる。マウスでボールを好きなところに動かし，左クリックでボールが動き出す。ボールが動いているとき，右クリックでもう一度ボールを置くところから始めることができ，プログラムから抜けるときは，ボールが動いているときに左クリック。フリッパーを動かすのはXF1キーである。

来月は，裏マップエディタでも作ろうかと思っている。　　　　（つづく）

サンプルの入力方法について

サンプルプログラムは，2つのリストに分かれている。1つはプログラム中で使うファイルのアーカイブを出力するプログラム，もう1つはフリッパーの付いたピンボールのデモである。リスト1は，見てわかるとおり2Kバイトほどのデータが並んでいる。これは今回使用するファイルを圧縮したものをそのままリストに落としたものだ。で，入力に間違いがあると当然正しいファイルが出力されない。そこで，リスト中には簡単なチェックサムを埋め込んでおいた。リスト1のファイルを実行したときに，入力ミスが見つかると間違えた行を表示し，実行をストップする。間違いがなければ"comp.lzh"というファイルを書き出す。そのあと，lha.xを使いファイルを解凍するのだが，その際はコマンドラインから，

lha e comp.lzh

とする。なお，lha.xをお持ちでない方は，過去の付録ディスクから探していただきたい。

あとはリスト2を入力後，コンパイルして実行するだけである。

リスト1

```
  1: #include"stdio.h"
  2: #include"class.h"
  3: UWORD  bm[1025] = {
  4:    0x2297,0x2D6C,0x6835,0x2D5F,0x0400,0x00F8,0x2900,0x0079,
  5:    0xB987,0x1C20,0x0109,0x7465,0x7374,0x312E,0x626D,0x7004,
  6:    0x0748,0x0000,0x0300,0x6CD7,0xD126,0xE7F9,0x5010,0x048A,
  7:    0x8900,0x1554,0xD00F,0x31F7,0xFDDD,0xD004,0x5451,0x59D5,
  8:    0xAD92,0xCBD1,0x7962,0xC997,0x39B6,0xD997,0xE0EE,0xB93B,
  9:    0xC7FE,0xF049,0xE607,0xDF7B,0x173B,0x3D52,0x595E,0xBD4F,
 10:    0xDCAF,0xE016,0xB1E2,0x0824,0x96B0,0xAAED,0xF353,0x4492,
 11:    0x4747,0x8B8A,0x28AB,0x7094,0x8484,0x7370,0x440A,0x16D6,
 12:    0x4022,0x13EF,0xC30E,0xEA0D,0x040B,0x44E3,0x098B,0x13C2,
 13:    0x7E0F,0xEA87,0x3D9B,0x303C,0x1830,0x181C,0x5C2C,0xA9C1,
 14:    0xDFBF,0x7F07,0x9D0F,0xA01D,0xEBD7,0xFA50,0xF282,0xBB76,
 15:    0xEF20,0x6849,0x43C6,0x15D7,0x7983,0x9394,0xE3A2,0x5C71,
 16:    0xCBBD,0x2CB9,0x6CD6,0xDBFB,0xD0F2,0x8369,0xA77F,0x343D,
 17:    0xB303,0xED98,0x1D6E,0x743A,0x4C14,0x1B3F,0xA077,0x0E0D,
 18:    0x91C2,0xD0FE,0x8112,0x4993,0x03B7,0x6BB2,0xC2C9,0xA692,
 19:    0x41DE,0xF56A,0xC3C2,0x3EFF,0x6941,0xDCC6,0x0E18,0x7903,
 20:    0x68A0,0xD898,0x1C3C,0xC1B6,0x7FCA,0x7941,0xC3E8,0x0DD3,
 21:    0x83B5,0xEC0E,0x3A1E,0xCD98,0xC1C4,0x79EA,0x0EA7,0x649C,
 22:    0x1FE6,0x0DF7,0xDB90,0x1AEB,0xFD81,0xB737,0xAA75,0x8CB6,
 23:    0x5EEE,0xD998,0x22CB,0x141F,0xB707,0x0E28,0xA4E0,0xE7A2,
 24:    0x64FE,0x427F,0x445A,0x4187,0xBB66,0x7071,0x4C03,0x1CA2,
 25:    0xCB08,0x8A28,0xA117,0x081C,0x1CFE,0x9F79,0xB93C,0x263F,
 26:    0x787C,0xD1FA,0x408E,0x3F8B,0xE381,0xD630,0x5A7C,0x16B3,
 27:    0x3C67,0xEC0F,0x59D6,0x759D,0x6759,0x3CEE,0xD797,0x984F,
 28:    0xF82D,0x8FF4,0x10FD,0xD0D0,0x24FC,0x0010,0x8504,0x1045,
 29:    0x56AD,0x5AC8,0x23D2,0xD5C4,0x1B0A,0x0A08,0x8E0E,0xBC73,
 30:    0x7811,0xEC25,0x8713,0xCF14,0x1D69,0xC1EF,0x0E3D,0x84F0,
 31:    0xE120,0x3A41,0x7D04,0x5E1C,0xBF20,0x3474,0x83DA,0x21D7,
 32:    0x5D25,0x1020,0xE53D,0xA0F5,0x8870,0x25E1,0x857A,0x0807,
 33:    0x438C,0x17C8,0x1E70,0xE506,0x1668,0xA0DE,0x0B20,0x66C7,
 34:    0x0B44,0x3C2A,0xBE60,0xE50A,0xD34D,0x102D,0xE203,0x43C3,
 35:    0x8374,0x0468,0x9002,0x6178,0x01A5,0xC41A,0x86D9,0x54D1,
 36:    0x965C,0x79E7,0xB959,0x5183,0x64F1,0x75D6,0x55EA,0xDCAC,
 37:    0xAC60,0xEE31,0x6AD2,0xEB2C,0xAA8A,0x5745,0x1F30,0x6A1C,
 38:    0x1DC6,0x1862,0xD050,0xCAA8,0x19B8,0xA48F,0xAD11,0x72E1,
 39:    0x812E,0x4A18,0x80D1,0x1FA2,0x1F90,0x736E,0xCCC0,0x0235,
 40:    0x0C98,0x0320,0x2F20,0xAA18,0x1DBB,0x76C8,0x0860,0xD3AF,
 41:    0x303F,0x10D9,0x419E,0x8698,0x1F78,0x5624,0xDA68,0xB432,
 42:    0x87B2,0xA607,0xB43B,0x126D,0x3564,0xD432,0x792C,0xAD41,
 43:    0x94A0,0x1D45,0xCBC1,0xAA13,0x83A9,0x0538,0x7528,0x043D,
 44:    0x956C,0x4B43,0x21FC,0x4CB4,0x34C1,0x1508,0xA66E,0x95D4,
 45:    0x1959,0x01B3,0x5270,0x9AF9,0x4767,0x0E33,0x527D,0xA738,
 46:    0x0390,0x2B54,0x7CC1,0x5F3B,0x2A70,0xADE9,0xF7FA,0xDE0C,
 47:    0x8174,0x0383,0x086D,0xCDBC,0x220E,0x6703,0x0405,0xFF8D,
 48:    0xE0DD,0xCE0A,0x5FE2,0xFC50,0xF3CF,0x707F,0x30CE,0x6EC1,
 49:    0x568B,0x003E,0x582E,0x60F6,0x518E,0x71C4,0x8FDC,0x1EA0,
 50:    0xCE7D,0x71B1,0xF707,0x942B,0x9ED9,0x61FB,0x8330,0x5EBA,
 51:    0x6E75,0x9D76,0x5730,0x5FF3,0xF424,0x3E4D,0x21E7,0x5A39,
 52:    0xF701,0x0121,0xBEC6,0xBE81,0x5E24,0x108C,0x0D04,0x7EBF,
 53:    0x7078,0x8F21,0x86A9,0xC1A6,0xAFD8,0x10C1,0x9288,0x0F7D,
 54:    0x24C7,0x86AF,0x00B7,0x492F,0xDF10,0xB09A,0xCC44,0x99E6,
 55:    0xA9C4,0x23A1,0x0814,0x0186,0xECC5,0x0401,0x3705,0x26C2,
 56:    0x240A,0x3DA7,0x841B,0xD523,0x5D35,0x14E4,0x11C8,0xC205,
 57:    0x1040,0x6704,0x409E,0x215B,0x2C28,0x41D9,0x83E9,0x84E4,
 58:    0x07E5,0xC2A4,0x024D,0x8483,0x76C0,0x0738,0x2FA0,0x4282,
 59:    0x1C18,0x40A2,0x10AD,0x9017,0x8F0A,0x1F65,0x6ED1,0xD161,
 60:    0x4421,0x450D,0xE1C2,0x85DA,0x5993,0x6C32,0xC0AA,0xD20B,
 61:    0x6848,0xF69A,0x64BA,0xEBDA,0xD20B,0xBECF,0x279F,0x26C3,
 62:    0x41A8,0x66A0,0x5F20,0x85F3,0xE502,0x0093,0xA416,0xD0B6,
 63:    0x8F0B,0xB4F0,0x698F,0xA41E,0xC08B,0x9E17,0x5D40,0xD42E,
 64:    0xBE80,0x5B02,0x2C48,0x5D74,0xC943,0x1FF6,0xDD5E,0x014F,
 65:    0x4249,0x4292,0x2BC8,0x2598,0xC129,0x202F,0x409D,0x9182,
 66:    0xC60F,0x9852,0x0257,0x302D,0x6169,0x0762,0x3059,0x41F1,
 67:    0x087E,0x0CA0,0x1E80,0xAB2D,0x4FCA,0xD0B5,0x3648,0x7C08,
 68:    0x4968,0x0BAB,0xE5AC,0xA6C9,0x48A1,0x3305,0x597D,0x4128,
 69:    0x0691,0xE660,0xABEB,0xFC27,0xEE04,0x017F,0x9E57,0x0D95,
 70:    0xFB81,0x67A2,0xA905,0x1EED,0xD483,0xCDC2,0xDF1F,0x450D,
 71:    0x40F2,0xD176,0xEA78,0x854B,0x684B,0x7DFE,0xC67A,0x81CD,
 72:    0x6833,0xE58F,0x1D10,0x3E02,0xB303,0x6B7E,0x453F,0x0A25,
 73:    0xBE39,0x384B,0x421F,0xFC80,0x5739,0xE56C,0x9502,0x5555,
 74:    0xDF5C,0xBF28,0x977F,0xBCD0,0xE3B2,0xCB2F,0x74CE,0x249B,
 75:    0x2818,0x661C,0x1F9C,0xF3CC,0x05EC,0x5FAD,0xB6C8,0x0C5F,
 76:    0xFEAC,0x4022,0x902D,0x6C68,0x352D,0xF401,0x0000,0xF829,
 77:    0x0000,0x79B9,0x871C,0x2001,0x0974,0x6573,0x7432,0x2E62,
 78:    0x6D70,0xD29F,0x4800,0x0001,0x0264,0x93AD,0xA57F,0xD782,
 79:    0x127B,0xDCC4,0x1042,0xA7CA,0xE352,0xCAD5,0xA04C,0x4A2D,
 80:    0x2422,0x0356,0x96A3,0x3E5B,0xECC7,0x51D0,0x755D,0x7371,
 81:    0xD173,0x1D47,0x87AF,0x04A7,0x4F00,0x3BC6,0x6D94,0x79FC,
 82:    0x117A,0xCC1C,0xC14D,0xCD91,0x854F,0xFEB2,0x051B,0xB86E,
 83:    0xFC40,0x1503,0x0880,0xA81E,0x1082,0x3C01,0x8D80,0xC7EC,
 84:    0x1C60,0x97A1,0xC220,0x7B76,0x662D,0x0973,0xF9A0,0xF568,
 85:    0x6BC0,0xC7E0,0x1AF6,0x55E0,0x9EBA,0x1FE4,0x19D5,0x9FA0,
 86:    0xD1E8,0x7FE8,0x1BDF,0x80EF,0xFBB7,0x7B70,0x38FE,0x4284,
 87:    0xF684,0xF206,0xF103,0xC217,0x7037,0xC2E9,0xF03E,0x3FB6,
 88:    0xFE7F,0xDF38,0x14C0,0xEF28,0x0F68,0x61BB,0x1D8E,0xC763,
 89:    0xB3DF,0x38D9,0x5F8E,0x8ABF,0x2E36,0x80E5,0x993E,0xF030,
 90:    0x4DAF,0xC40C,0x33AF,0xD60D,0x5799,0xA2D3,0x440D,0x1A46,
 91:    0x8A5F,0x103F,0xF06A,0xEA8D,0x6DEA,0x7004,0xF406,0x6E1B,
 92:    0x2A20,0x6635,0xBA1C,0xE7DA,0x121A,0x1A10,0x39C2,0x84B4,
 93:    0xBD20,0x79C3,0x867A,0xA2CD,0xC399,0x96A8,0xA40E,0x40E0,
 94:    0xCAC4,0xAB8D,0x8A4A,0xA3CE,0x125A,0xBF4D,0xC85F,0x9AB8,
 95:    0xA8E4,0xB865,0x95F1,0xF27E,0x7859,0x5E10,0x583D,0xC077,
 96:    0x5A63,0x183B,0x9658,0xD36D,0xF092,0x0FF5,0x03C3,0x6DDA,
 97:    0x743C,0x9710,0xF306,0x90C1,0xA20A,0xC40F,0xAC28,0x698B,
 98:    0x9103,0x8830,0x6C8D,0x9103,0x8028,0x5530,0x6207,0x782B,
 99:    0xA9B8,0xEC75,0x95F0,0x7D50,0xDB65,0x4A83,0x4154,0x9FF8,
100:    0x1BB6,0x5868,0x365A,0x531F,0x58BE,0xC0BD,0x1B24,0x77DC,
101:    0x7E0B,0x9110,0x3352,0x0F5E,0x8A4D,0x62C8,0x199B,0xE076,
102:    0x709A,0x3B1A,0x4819,0x8819,0x95E6,0xD6D0,0x90F5,0xF0B7,
103:    0x8502,0x5B70,0x2C1B,0x0223,0xB095,0x4145,0x9723,0xD410,
104:    0xF504,0x4DA4,0x9210,0xF684,0xEDB8,0xEE50,0x935D,0x8A50,
105:    0x935E,0xA327,0x9C75,0x847B,0xB2FE,0x2CF5,0xEE7A,0xC25C,
106:    0x81C2,0xB81E,0x7AB9,0xB527,0xE077,0xA6B8,0x88E0,0x74D1,
107:    0x5373,0x4BC0,0xE8BD,0x1738,0x3903,0x281C,0x6156,0x0E30,
108:    0xCB05,0xD214,0xDC81,0x0385,0xB74C,0x165B,0x768B,0x2BE2,
109:    0xB6E8,0x36DB,0x5B80,0xBE18,0x3FAA,0xC424,0xAB2D,0x6C68,
110:    0x352D,0x3F01,0x0000,0x0008,0x0000,0x6181,0x861C,0x2001,
111:    0x0B70,0x6164,0x2662,0x616C,0x6C2E,0x5350,0xAB96,0x4800,
112:    0x0000,0xE55A,0x76DA,0x36B0,0x3F65,0x8BD0,0x30D6,0x428E,
113:    0xD1DA,0x0E63,0xA5D6,0x8C8B,0x7AC6,0x41B6,0x3E50,0xD746,
114:    0x51D6,0xF847,0xFC8F,0xFC1F,0xEB75,0xEB4C,0x6A18,0xC670,
115:    0x985D,0x0558,0x0E50,0xC298,0x14B0,0x57C2,0xC718,0xBB56,
116:    0xDA74,0x041A,0xB830,0x6668,0x9F9C,0xC97D,0xB959,0xE66F,
117:    0x95DD,0x495F,0xE23D,0xA884,0xEBE9,0xF054,0xAB7B,0x9517,
118:    0xAC0C,0x8782,0x3C50,0x3CC0,0x994D,0xD7CE,0xEA6B,0x7447,
119:    0x7906,0x8D23,0xFCD0,0x12FB,0x21F2,0xD7FF,0x069E,0x802E,
120:    0x613C,0xA803,0xF63F,0xE716,0xF2DC,0xE3E5,0x534D,0x90FE,
121:    0x5B68,0x3C79,0x6E0F,0xF207,0x5016,0x75BF,0x9D18,0xB7AE,
122:    0x7E4B,0x38E7,0xFD94,0xF933,0xEC1F,0xF967,0xFE9D,0x03C6,
123:    0x65EA,0xC6BF,0x7F01,0x64BD,0x0BF1,0x8BEC,0xC77A,0xAAFF,
124:    0xF5FB,0x97E7,0x93C4,0x9A2B,0xC7FA,0x743C,0xFCE1,0x7ABC,
125:    0x55A2,0x9CD3,0x3D64,0xBE88,0xBE2A,0x6FD3,0x3868,0x8BAC,
126:    0xF35F,0x3928,0xCDE5,0x70E0,0x6304,0x6396,0x48AC,0x73C5,
```

```
127:    0x7F72,0x2BF0,0xB11C,0x305A,0x4C18,0x3892,0x9C3B,0xEDBC,
128:    0xEEED,0xD75B,0xB171,0xA745,0xB4BC,0x1872,0x102D,0x472B,
129:    0xDF0A,0xBE2D,0xB207,0x5C2C,0x881C,0xC212,0xE3F6,0x5298,
130:    0x383B,0x9EE9,0xBD05,0xC4F5,0x3D75,0x3D8E,0xBAFD,0xE69B,
131:    0x7F8F,0x91A7,0xACA6,0x2ED1,0x148C,0x893A,0x00E6,0x53D8,
132:    0x0000 };
133: /* この部分はチェックサムですので、正確に入力して下さい */
134: UBYTE  sum[64] = {
135:    0xFC,0x41,0xFB,0xE1,0x90,0xC0,0xA1,0xDC,0xD1,0xE3,0x83,
136:    0x71,0x06,0xFA,0x49,0x19,0xFE,0x19,0x47,0x2A,0x53,0x59,
137:    0x9E,0x1A,0x12,0x61,0x1B,0x53,0xCF,0xC7,0xAF,0xC6,0xBB,
138:    0x26,0xB1,0xA8,0x7D,0x75,0x48,0xFD,0xE0,0xDE,0xD1,0x5D,
139:    0x60,0x2C,0x73,0x4C,0xB2,0x92,0xB9,0x5D,0xD8,0x11,0xF7,
140:    0xDA,0xBB,0xD2,0x4E,0xC9,0xA6,0x54,0x76,0xC9 };
141: main()
142: {
143:  FILE        *fp;
144:  int         i,j,s = 0,ps;
145:       /* 入力ミスがないかどうかチェックする */
146:       for( i = 0; i != 64; i ++ )
147:       {
148:           s = 0;
149:           for( j = 0; j != 16; j++ )
150:           {
151:               if( j & 1 )
152:                   ps = (bm[i*16+j]&0xffff)^0xffff;
153:               else
154:                   ps = (bm[i*16+j]&0xffff);
155:               s += ps;
156:               s += s >> 8;
157:               s &= 0xff;
158:           }
159:           if( s != sum[i] )
160:           {
161:               printf( "%d、または%d行",i*2+4,i*2+5 );
162:               printf( "に間違いがあるようです。" );
163:               exit( 0 );
164:           }
165:       }
166:       fp = fopen( "comp.lzh","wb" );
167:       for( i = 0; i != 1029; i++ )
168:           putw( bm[i],fp );
169:       fclose( fp );
170: }
```

リスト2

```
  1: #include "stdio.h"
  2: #include "iocslib.h"
  3: #include "basic.h"
  4: #include "graph.h"
  5: #include "class.h"
  6:
  7: typedef  struct  b_parm {
  8:    int  x,y;
  9:    int  dx,dy,cx,cy;
 10:    int  ccx,ccy,drx,dry,pl;
 11:    } b_parm,*b_parmPtr;
 12:
 13: typedef  struct  r_parm {
 14:    int  x,y,dx,dy,cx,cy,f,drx,dry;
 15:    int  dxx,dyy,ddx,ddy,bx,by,ex,ey;
 16:    } r_parm,*r_parmPtr;
 17:
 18: b_parm  bp;
 19: UBYTE   bmap[512][512];
 20: UBYTE   pad[2][158][68];
 21: int     si[65],co[65],s2[65],c2[65];
 22: UBYTE   k[16];
 23: int     pox[10] = {
 24:         0, 0, 0,-1,-1,-1, 1, 1, 1 }
 25:        ,poy[10] = {
 26:         0, 1,-1, 0, 1,-1, 0, 1,-1 };
 27: UWORD   pat[16][64];
 28: struct  PD_PAT {
 29:    int   ox,oy,hv;
 30:    int   sp[6];
 31:    } l_paddle[5] = {
 32:    86,459,0,
 33:    9,10,11, 12,13,14,
 34:    86,459,0,
 35:    3,4,5, 6,7,8,
 36:    86,459,0,
 37:    0,1,2, -1,-1,-1,
 38:    86,442,2,
 39:    6,7,8, 3,4,5,
 40:    86,442,2,
 41:    12,13,14, 9,10,11 };
 42:
 43: struct  P_PARM  {
 44:    int   lps,rps,lpv,rpv;
 45: } pd;
 46:
 47: UWORD  pal[16] = {
 48:  0x0000,0x18C6,0x318C,0x4A52,0x6318,0x8420,0xA528,0xCE72,
 49:  0xFFFE,0x4000,0xCE72,0x4010,0x0001,0x2D8C,0x5ECB,0x9FD5 };
 50:
 51: int   move_ball();
 52: void  feeling();
 53: void  flipper();
 54: void  reflect();
 55: void  calc_parm();
 56: void  locate_ball();
 57: void  put_ball();
 58: void  draw_screen();
 59: void  create_map();
 60: void  store_sctab();
 61: void  set_sprite();
 62:
 63: main()
 64: {
 65:  int  i = 0,j;
 66:     OS_CUROF();
 67:     draw_screen();
 68:     store_sctab();
 69:     create_map();
 70:     set_sprite();
 71:     while( i == 0 )
 72:     {
 73:       put_paddle( &l_paddle[0],0 );
 74:       locate_ball( &bp );
 75:       main_loop();
 76:       msstat( &j,&j,&i,&j );
 77:     }
 78:     OS_CURON();
 79:     exit(0);
 80: }
 81:
 82: void  main_loop()
 83: /* メインループ */
 84:
 85:  {
 86:  int  i = 0,j = 0,k,f = 0;
 87:  int  t1,t2,t3;
 88:    bp.dx = 0;
 89:    bp.dy = 0;
 90:    bp.drx = 0;
 91:    bp.dry = 0;
 92:    bp.cx = 0;
 93:    bp.cy = 0;
 94:    bp.ccx = 0;
 95:    bp.ccy = 0;
 96:    bp.pl = 0;
 97:    pd.lps = 0;
 98:    pd.lpv = 0;
 99:    while( i == 0 && j == 0 && bp.y < 505 )
100:    {
101:      inkey();
102:      do_paddle();
103:      move_ball( &bp );
104:      bp.dy += 0x2000; /* この値を大きくすると */
105:      put_ball( bp.x,bp.y ); /* ボールが機敏に動く */
106:      put_paddle( &l_paddle[pd.lps],0 );
107:      bp.dx = bp.dx*127/128;
108:      bp.dy = bp.dy*127/128;
109:      msstat( &k,&k,&i,&j );
110:    }
111:  }
112:
113: void  inkey()
114:
115:  {
116:  int  i;
117:    for( i = 0; i < 16; i++ )
118:      k[i] = BITSNS( i );
119:  }
120:
121: int  move_ball( bp )
122: b_parmPtr  bp;
123:
124:  {
125:  int      i,r = 0,r2,rf;
126:  r_parm  rp;
127:    if( (*bp).pl != 0 )
128:      (*bp).pl--;
129:    (*bp).cx += (*bp).dx;
130:    (*bp).cy += (*bp).dy;
131:    if( (*bp).cx > 0 )
132:      rp.dx = (*bp).cx >> 16;
133:    else
134:      rp.dx = (-(*bp).cx) >> 16;
135:    if( (*bp).cy > 0 )
136:      rp.dy = (*bp).cy >> 16;
137:    else
138:      rp.dy = (-(*bp).cy) >> 16;
139:    (*bp).cx %= 65536;
140:    (*bp).cy %= 65536;
141:    if( rp.dx == 0 && rp.dy == 0 )
142:    {
143:      return( 0 );
144:    }
145:    calc_parm( bp,&rp );
146:    rp.cx = rp.f / 2;
147:    rp.cy = rp.f / 2;
148:    rp.cx = (*bp).ccx;
149:    rp.cy = (*bp).ccy;
150:    while( rp.ex > 0 || rp.ey > 0 )
151:    {
152:      rf = 0;
153:      rp.bx = rp.x;
154:      rp.by = rp.y;
155:      rp.ddx = rp.ex;
156:      rp.ddy = rp.ey;
157:      rp.cx += rp.dx;
158:      if( rp.cx >= rp.f )
159:      {
160:        rp.x += rp.drx;
161:        rp.ex--;
162:        rp.cx -= rp.f;
163:        r = bmap[rp.x][rp.y];
164:        if( r != 255 )
165:        {
166:          rf = 1;
167:        }
168:      }
```


▶定期購読よろしくお願いいたします。社会人はつらいッす。　寺田　亮治(22)栃木県

```
169:       rp.cy += rp.dy;
170:       if( rp.cy >= rp.f )
171:       {
172:         rp.y += rp.dry;
173:         rp.ey--;
174:         rp.cy -= rp.f;
175:         r2 = bmap[rp.x][rp.y];
176:         if( r2 != 255 )
177:         {
178:           rf = 1;
179:           r = r2;
180:         }
181:         else
182:         {
183:           rf = 0;
184:         }
185:       }
186:       if( rf != 0 )
187:       {
188:         feeling( bp,&rp,r );
189:       }
190:       if( rp.x > 75 && rp.x < 153
191:        && rp.y > 389 && rp.y < 489 )
192:         l_pad( bp,&rp );
193:     }
194:     (*bp).x = rp.x;
195:     (*bp).y = rp.y;
196:     (*bp).ccx = rp.cx;
197:     (*bp).ccy = rp.cy;
198:     return( rf );
199:   }
200: 
201: void  do_paddle()
202: 
203:   {
204:     if( pd.lpv == 0 && pd.lps != 4 )
205:     {
206:       if( k[10] & 32 )
207:       {
208:         pd.lpv = 1;
209:       }
210:     }
211:     else
212:     {
213:       if( pd.lps == 4 )
214:       {
215:         pd.lpv = 0;
216:         if( !(k[10] & 32) )
217:         {
218:           pd.lpv = -1;
219:           pd.lps = 3;
220:         }
221:       }
222:       else
223:       {
224:         pd.lps += pd.lpv;
225:       }
226:       if( pd.lps == 0 && pd.lpv == -1 )
227:       {
228:         pd.lpv = 0;
229:       }
230:     }
231:   }
232: 
233: void  feeling( bp,rp,r )
234: b_parmPtr  bp;
235: r_parmPtr  rp;
236: int  r;
237: 
238:   {
239:     if( r < 65 )
240:       reflect( bp,rp,r );
241:     else
242:     {
243:       if( r == 65 || r == 66 )
244:         flipper( bp,rp,r );
245:       if( r == 67 )
246:         circ( bp,rp,r );
247:     }
248:   }
249: 
250: void  l_pad( bp,rp )
251: b_parmPtr  bp;
252: r_parmPtr  rp;
253: 
254:   {
255:   int  j = 0,x,y,r1 = 0,r2 = 0,r = 0,i = 0,dx = 0,dy = 0,d;
256:   int  ff = 0,ii = 1,jj = 0,i1 = 0,i2 = 0,pdd = -1;
257:     x = (*rp).x - 74;
258:     y = (*rp).y - 447;
259:     if( pd.lpv != 0 && (*bp).pl == 0 )
260:     {
261:       i2 = fll( bp,x,y,pd.lps,1 );
262:       if( i2 == 0x29 && pd.lps - pd.lpv >= 0
263:        && pd.lps - pd.lpv != 5 )
264:         i1 = fll( bp,x,y,pd.lps - pd.lpv,1 );
265:       if( i2 != 0 )
266:       {
267:         i = i2;
268:       }
269:       if( i1 != 0 )
270:       {
271:         i = i1;
272:         i2 = 0;
273:       }
274:       if( i2 != 0 )
275:       {
276:         r = fll( bp,x,y,pd.lps,0 );
277:         pdd = pd.lps;
278:       }
279:       else
280:       {
281:         r = fll( bp,x,y,pd.lps - pd.lpv,0 );
282:         pdd = pd.lps - pd.lpv;
283:       }
284:       j = 1;
285:     }
286:     else
287:     {
288:       r = fll( bp,x,y,pd.lps,0 );
289:       i = fll( bp,x,y,pd.lps,1 );
290:       if( (*bp).pl )
291:       {
292:         if( i != 0x28 && i != 0x29 )
293:           r = 0;
294:       }
295:       if( i != 0x29 && i != 0x27 )
296:         i = 0x28;
297:     }
298:     if( r != 0 && i != 0 )
299:     {
300:       (*rp).x = (*rp).bx;
301:       (*rp).y = (*rp).by;
302:       (*rp).ex = (*rp).ddx;
303:       (*rp).ey = (*rp).ddy;
304:       r--;
305:       r2 = r * 1.7777;
306:       reflect( bp,rp,r2 );
307:       if( j != 0 && i != 0x28 )
308:       {
309:         r = r2;
310:         if( i == 0x26 )
311:         {
312:           if( pdd > 1 )
313:           {
314:             dx = c2[r];
315:             dy = s2[r];
316:           }
317:           else
318:           {
319:             dx = -c2[r];
320:             dy = -s2[r];
321:           }
322:         }
323:         if( i == 0x25 )
324:         {
325:           if( pdd > 2 )
326:           {
327:             dx = -c2[r];
328:             dy = -s2[r];
329:           }
330:           else
331:           {
332:             dx = c2[r];
333:             dy = s2[r];
334:           }
335:         }
336:         if( i != 0x27 && i != 0x29 )
337:         {
338:           d = x;
339:           if( d > 0 )
340:           {
341:             d += 140;
342:       /* この値を大きくするとボールが高く弾かれる */
343:             (*bp).dx /= 4;
344:             (*bp).dy /= 4;
345:             (*bp).dx += dx*d;
346:             (*bp).dy += dy*d;
347:             (*bp).cx += (*bp).dx;
348:             (*bp).cy += (*bp).dy;
349:             if( (*bp).cx > 0 )
350:               (*rp).dx = (*bp).cx >> 16;
351:             else
352:               (*rp).dx = (-(*bp).cx) >> 16;
353:             if( (*bp).cy > 0 )
354:               (*rp).dy = (*bp).cy >> 16;
355:             else
356:               (*rp).dy = (-(*bp).cy) >> 16;
357:             (*bp).cx %= 65535;
358:             (*bp).cy %= 65535;
359:             calc_parm( bp,rp );
360:             (*bp).pl = 4;
361:           }
362:         }
363:         else
364:           if( i == 0x27 )
365:           {
366:             (*bp).pl = 4;
367:           }
368:       }
369:     }
370:   }
371: 
372: int  fll( bp,x,y,pp,o )
373: 
374: b_parmPtr  bp;
375: int  x,y,pp,o;
376: 
377:   {
378:   int  i = 0,k = 0,l,di = 0,d2 = 0,tt;
379:     if( pd.lpv == 1 )
380:     {
381:       di = 0x26;
382:       d2 = 0x25;
383:     }
384:     if( pd.lpv == -1 )
385:     {
386:       di = 0x25;
387:       d2 = 0x26;
388:     }
389:     if( pp > 2 )
390:     {
```

▶「PUSH BON!」はよかった。ゆっくりできてそこそこ頭を使い，かわいいキャラで市販ソフトよりもよかった。ぜひ次号でデータ集を……。　古味　貴徳(20)千葉県

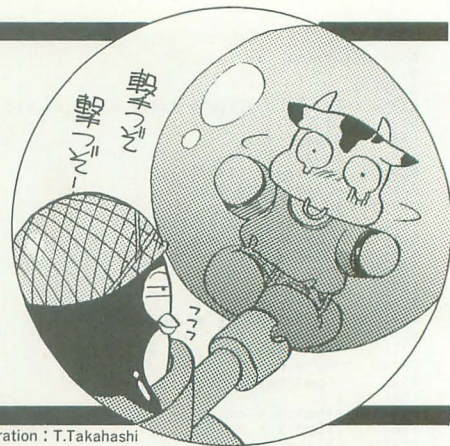
```
391:       k = 1;
392:       pp = 4 - pp;
393:       y = 41 - y;
394:     }
395:     if( y < 1 || y > 52 )
396:       return( 0 );
397:     y += (2-pp)*52;
398:     if( x > 67 || x < 1 || y > 157 || y < 1 )
399:       return( 0 );
400:     for( l = 0; l != 9; l++ )
401:     {
402:       i = pad[o][y+poy[l]][x+pox[l]];
403:       if( i != 0 )
404:       {
405:         break;
406:       }
407:     }
408:     if( k == 1 )
409:     {
410:       if( o != 1 )
411:       {
412:         if( i != 0 )
413:         {
414:           i = 36 - (i-1);
415:           i %= 36;
416:           i++;
417:         }
418:       }
419:       else
420:       {
421:         if( i == 0x25 )
422:           i = 0x26;
423:         else
424:           if( i == 0x26 )
425:             i = 0x25;
426:       }
427:     }
428:     if( i == di && di != 0 )
429:     {
430:       if( pd.lpv == 1 )
431:         i = d2;
432:       else
433:         i = 0x29;
434:     }
435:     return( i );
436:   }
437:
438: void  flipper( bp,rp,r )
439: b_parmPtr  bp;
440: r_parmPtr  rp;
441: int  r;
442:
443:   {
444:     r = 48;
445:     reflect( bp,rp,r );
446:     (*bp).dx -= c2[r] * 90;
447:     (*bp).dy -= s2[r] * 90;
448:     (*bp).cx += (*bp).dx;
449:     (*bp).cy += (*bp).dy;
450:     if( (*bp).cx > 0 )
451:       (*rp).dx = (*bp).cx >> 16;
452:     else
453:       (*rp).dx = (-(*bp).cx) >> 16;
454:     if( (*bp).cy > 0 )
455:       (*rp).dy = (*bp).cy >> 16;
456:     else
457:       (*rp).dy = (-(*bp).cy) >> 16;
458:     (*bp).cx %= 65535;
459:     (*bp).cy %= 65535;
460:     calc_parm( bp,rp );
461:   }
462:
463: void  circ( bp,rp,r )
464: b_parmPtr  bp;
465: r_parmPtr  rp;
466: int  r;
467:
468:   {
469:   int  r2,x;
470:     r2 = bmap[(*rp).x+255][(*rp).y];
471:     reflect( bp,rp,r2 );
472:     x = (*rp).x;
473:     if( r2 == 0 )
474:     {
475:       if( (*rp).y > 90 )
476:         x = 160;
477:       else
478:         x = 100;
479:     }
480:     if( x > 152 )
481:     {
482:       (*bp).dx += c2[r2] * 64;
483:       (*bp).dy += s2[r2] * 64;
484:     }
485:     else
486:     {
487:       (*bp).dx -= c2[r2] * 64;
488:       (*bp).dy -= s2[r2] * 64;
489:     }
490:     (*bp).cx += (*bp).dx;
491:     (*bp).cy += (*bp).dy;
492:     if( (*bp).cx > 0 )
493:       (*rp).dx = (*bp).cx >> 16;
494:     else
495:       (*rp).dx = (-(*bp).cx) >> 16;
496:     if( (*bp).cy > 0 )
497:       (*rp).dy = (*bp).cy >> 16;
498:     else
499:       (*rp).dy = (-(*bp).cy) >> 16;
500:     (*bp).cx %= 65535;
501:     (*bp).cy %= 65535;
502:     calc_parm( bp,rp );
503:   }
504:
505: void  reflect( bp,rp,r )
506: /*  反射の処理  */
507: b_parmPtr  bp;
508: r_parmPtr  rp;
509: int  r;
510:
511:   {
512:   int     i,j,k,dx,dy,df,dxx,dyy,ddx,ddy;
513:   int     r1,r2,dr1,dr2,dr3;
514:   double  f1,f2,f3,f4,f5,f6;
515: /*  r--;*/
516:     ddx = (*bp).drx;
517:     ddy = (*bp).dry;
518:     dx = (*bp).dx;
519:     dy = (*bp).dy;
520:     /*  X方向の反射  */
521:     i = dx * co[r];
522:     j = dy * si[r];
523:     k = i + j;
524:     if( k > 0 )
525:     {
526:       (*bp).drx = 1;
527:       dxx = k / 4096;
528:     }
529:     else
530:     {
531:       (*bp).drx = -1;
532:       dxx = k / 4096;
533:     }
534:     /*  Y方向の反射  */
535:     i = dx * si[r];
536:     j = dy * co[r];
537:     k = i - j;
538:     if( k > 0 )
539:     {
540:       (*bp).dry = 1;
541:       dyy = k / 4096;
542:     }
543:     else
544:     {
545:       (*bp).dry = -1;
546:       dyy = k / 4096;
547:     }
548:     f1 = (double)dxx;
549:     f2 = (double)dyy;
550:     f4 = -(double)dx;
551:     f5 = -(double)dy;
552:     f3 = sqrt(f1*f1 + f2*f2)*sqrt(f4*f4 + f5*f5);
553:     f6 = (1+(f1*f4+f2*f5)/f3)/3;
554:     (*bp).dx = dxx - dxx*f6;
555:     (*bp).dy = dyy - dyy*f6;
556:     /*  反射後ボールがどれだけ動くかを計算  */
557:     dx = (*bp).dx;
558:     dy = (*bp).dy;
559:     r1 = (*rp).dxx*(*rp).dxx + (*rp).dyy*(*rp).dyy;
560:     r2 = (*rp).dx*(*rp).dx + (*rp).dy*(*rp).dy;
561:     if( r1 != 0 )
562:     {
563:       (*bp).cx += (dx*r2/r1);
564:       (*bp).cy += (dy*r2/r1);
565:       if( (*bp).cx > 0 )
566:         (*rp).dx = (*bp).cx / 65536;
567:       else
568:         (*rp).dx = (-(*bp).cx) / 65536;
569:       if( (*bp).cy > 0 )
570:         (*rp).dy = (*bp).cy / 65536;
571:       else
572:         (*rp).dy = (-(*bp).cy) / 65536;
573:       (*bp).cx %= 65535;
574:       (*bp).cy %= 65535;
575:     }
576:     else
577:     {
578:       (*rp).dx = 0;
579:       (*rp).dy = 0;
580:     }
581:     calc_parm( bp,rp );
582:   }
583:
584: void  calc_parm( bp,rp )
585: /*  移動用パラメーターの算出  */
586: b_parmPtr  bp;
587: r_parmPtr  rp;
588:
589:   {
590:     if( (*bp).dx > 0 )
591:       (*bp).drx = 1;
592:     else
593:       (*bp).drx = -1;
594:     if( (*bp).dy > 0 )
595:       (*bp).dry = 1;
596:     else
597:       (*bp).dry = -1;
598:     (*rp).x = (*bp).x;
599:     (*rp).y = (*bp).y;
600:     (*rp).drx = (*bp).drx;
601:     (*rp).dry = (*bp).dry;
602:     if( (*rp).dx > (*rp).dy )
603:       (*rp).f = (*rp).dx;
604:     else
605:       (*rp).f = (*rp).dy;
606:     (*rp).ex = (*rp).dx;
607:     (*rp).ey = (*rp).dy;
608:     (*rp).dxx = (*rp).dx;
609:     (*rp).dyy = (*rp).dy;
610:   }
611:
612: void  locate_ball( bp )
```


▶楽しみにしていた計測技研の広告がない。040turboを買うつもりで詳細を待っていたのに。
大井　祐樹(10)神奈川県

```
613: b_parmPtr  bp;
614:
615:  {
616:  int  i = 0,j,k,x = 128,y = 128,mx,my;
617:    SP_ON();
618:    mouse( 4 );
619:    mouse( 2 );
620:    setmspos( 128,20 );
621:    while( i == 0 )
622:    {
623:      msstat( &j,&j,&i,&j );
624:      mspos( &mx,&my );
625:      if( point( mx,my ) != 4 )
626:      {
627:        mx = x;
628:        my = y;
629:        setmspos( mx,my );
630:      }
631:      x = mx;
632:      y = my;
633:      put_ball( mx,my );
634:    }
635:    (*bp).x = mx;
636:    (*bp).y = my;
637:    while( i != 0 )
638:      msstat( &j,&j,&i,&j );
639:  }
640:
641: void  put_ball( x,y )
642: int  x,y;
643:
644:  {
645:  int  i = 1 << 8;
646:    SP_REGST( 15,x+8,y+6,(i+15),3 );
647:  }
648:
649: void  put_paddle( p )
650:
651: struct PD_PAT  *p;
652:
653:  {
654:  int      x,y,i,j,hv,sp = 0;
655:  unsigned int   ad = 0xeb0000;
656:    x = (*p).ox+16;
657:    y = (*p).oy+16;
658:    for( i = 0; i != 2; i++ )
659:    {
660:      x = (*p).ox+16;
661:      for( j = 0; j != 3; j++ )
662:      {
663:        if( (*p).sp[i*3+j] != -1 )
664:        {
665:          B_WPOKE( ad  ,x );
666:          B_WPOKE( ad+2,y );
667:          B_WPOKE( ad+4,(*p).sp[sp] );
668:          B_WPOKE( ad+6,3 );
669:        }
670:        x += 16;
671:        ad += 8;
672:        sp++;
673:      }
674:      y += 16;
675:    }
676:  }
677:
678: void  draw_screen()
679:
680:  {
681:    screen( 1,1,1,1 );
682:    palet( 0,0 );
683:    palet( 1,65535 );
684:    palet( 2,rgb(  5,31,31 ) );
685:    palet( 3,rgb(  5, 8,31 ) );
686:    palet( 4,rgb(  5, 8,31 ) );
687:    palet( 5,rgb( 31,31, 2 ) );
688:    fill( 0,0,300,511,2 );
689:    circle( 60,60,50,3,0,360,255 );
690:    paint( 60,60,3 );
691:    circle( 239,60,50,3,0,360,255 );
692:    paint( 239,60,3 );
693:    fill( 70,11,231,448,3 );
694:    fill( 10,60,289,388,3 );
695:    fill( 60,21,239,389,4 );
696:    fill( 18,60,281,389,4 );
697:    line( 289,389,230,448,5,0xffff );
698:    line( 11,389,70,448,3,0xffff );
699:    line( 288,389,229,448,3,0xffff );
700:    paint( 12,389,3 );
701:    paint( 287,389,3 );
702:    fill( 85,448,229,511,3 );
703:    line( 230,448,152,487,2,0xffff );
704:    line( 152,511,152,487,2,0xffff );
705:    paint( 153,511,2 );
706:    circle( 152,90,35,3,0,360,255 );
707:    paint( 152,90,3 );
708:    circle( 152,90,25,5,0,360,255 );
709:    paint( 172,90,5 );
710:  }
711:
712: void  create_map()
713: /* 接触情報マップを作成する */
714:
715:  {
716:  int      i,x,y,i1,i2;
717:  double  a,b,p = 3.14159;
718:    for( x = 0; x != 512; x++ )
719:      for( y = 0; y != 512; y++ )
720:        bmap[x][y] = 255;
721:    for( i = 0; i != 91; i++ )
722:    {
723:      a = (double)i/(double)180;
724:      x = (int)(cos(a*p)*43+0.5);
725:      y = (int)(sin(a*p)*43+0.5);
726:      b = (double)i/2.8125;
727:      bm( 60-x,62-y,(int)(32+b) % 64);
728:      bm( 239+x,62-y,(int)(32-b) );
729:      x = (int)(cos(a*p)*33+0.5);
730:      y = (int)(sin(a*p)*33+0.5);
731:      bm( 152-x,90-y,67 );
732:      bm( 152+x,90-y,67 );
733:      bm( 152-x,90+y,67 );
734:      bm( 152+x,90+y,67 );
735:      bm( 407-x,90-y,(int)(32+b) % 64 );
736:      bm( 407+x,90-y,(int)(32-b) );
737:      bm( 407-x,90+y,(int)(32-b) % 64 );
738:      bm( 407+x,90+y,(int)(32+b) );
739:    }
740:    for( i = 0; i != 92; i++ )
741:    {
742:      if( i > 10 && i < 66 )
743:      {
744:        i1 = 16;
745:        i2 = 66;
746:      }
747:      else
748:      {
749:        i1 = 16;
750:        i2 = 48;
751:      }
752:      bmap[9+i][381+i] = i1;
753:      bmap[10+i][381+i] = i1;
754:      bmap[289-i][381+i] = i2;
755:      bmap[288-i][381+i] = i2;
756:      bmap[152+i][479-i/2] = 52;
757:      bmap[153+i][479-i/2] = 52;
758:    }
759:    /*.平らな壁の接触情報 */
760:    for( i = 0; i != 71; i++ )
761:    {
762:      bmap[90-i][444] = 5;
763:      bmap[90][445+i] = 32;
764:    }
765:    for( i = 59; i != 241; i++ )
766:      bmap[i][20] = 0;
767:    for( i = 487; i != 511; i++ )
768:      bmap[144][i] = 32;
769:    for( i = 59; i != 389; i++ )
770:    {
771:      bmap[17][i] = 32;
772:      bmap[282][i] = 32;
773:    }
774:  }
775:
776: bm( x,y,r )
777: int  x,y,r;
778:
779:  {
780:    bmap[x][y] = r;
781:    bmap[x-1][y] = r;
782:  }
783:
784: void  store_sctab()
785:
786:  {
787:  int      i;
788:  double  a,p = 3.1415;
789:    for( i = 0; i != 64; i++ )
790:    {
791:      a = (double)i/(double)32;
792:      si[i] = sin( a*p )*(double)4096;
793:      co[i] = cos( a*p )*(double)4096;
794:    }
795:    for( i = 0; i != 64; i++ )
796:    {
797:      a = (double)i/(double)64;
798:      s2[i] = -cos( a*p )*(double)4096;
799:      c2[i] = sin( a*p )*(double)4096;
800:    }
801:  }
802:
803: void  set_sprite()
804:
805:  {
806:  int   i,j;
807:  FILE  *fp;
808:    SP_INIT();
809:    fp = fopen( "pad&ball.sp","rb" );
810:    for( i = 0; i != 16; i++ )
811:    {
812:      fread( &pat[i],2,64,fp );
813:      SP_DEFCG( i,1,&pat[i] );
814:    }
815:    fclose( fp );
816:    for( i = 0; i != 16; i++ )
817:      SPALET( i,1,pal[i] );
818:    for( i = 0; i != 10; i++ )
819:    {
820:      for( j = 0; j != 6; j++ )
821:      {
822:        l_paddle[i].sp[j] +=
823:          (l_paddle[i].hv << 14) + (1 << 8);
824:      }
825:    }
826:    fp = fopen( "test1.bmp","rb" );
827:    fread( &pad[0][0][0],4,2686,fp );
828:    fclose( fp );
829:    fp = fopen( "test2.bmp","rb" );
830:    fread( &pad[1][0][0],4,2686,fp );
831:    fclose( fp );
832:  }
833:
```

ショートプロってなんだっけ!?

Komura Satoshi 古村　聡

今月は3本ともBASICのプログラム。だからみんなが大丈夫……というわけではなく，そのうち1本はSX-BASIC用のプログラムなのです。短いものから長いものまで用意してありますので，3本のゲームをゆっくりと楽しんでみてくださいね。

illustration：T.Takahashi

早いもんですねぇ。この「ショートプロ」も今回で57回を数えました。なんだかんだであと2回でこの連載も5年です。「絶対Z80's Barよりも長く続けてやるんだぁ！」と吠えていたら本当にむこうは終わって新連載に入っちゃってるし。

5年っていえば，当時小学生だった人も早い人では社会人，もしかしたら大学生だった人はいまでも大学生(笑)！　私はまっとうな社会人になれたけど。それにハイハイもできなかった子だっていまじゃセーラームーンのバッグにセーラームーンの靴，ムーンクリスタルをもって公園で遊んでますぜ，旦那。いや，このあいだ本当にそういうセラムングッズだらけの子供を公園で見ちゃいました。「おいらにもセラムングッズひとつ分けて～」と叫んでしまいそうに……ってそういう話じゃないんですよね。でも，UFOキャッチャーでムーン，マーキュリー，マーズのスリッパを連チャンで取ったから……全部片っぽだけだけど。どうだ，うらやましいだろう，えっへん(しかし，5歳の子供と同じものをほしがるオトナっていったい……)。

なんか私の書いていることや，そもそも人間の中味もほとんど変わっていないような気がします。むー，こんなことじゃいかんかもしれんな。よし，今回はショートプロとはいったいなにかをじっくり考えていくのだ！

ちびちび爆発タンクなのだ！

てことで気合いを入れてまず1本。今月，最初のプログラムは短いけど，2人で燃えるB_TANK.BASです。どうぞっ。

B_TANK.BAS for X680x0
(X-BASIC，要ジョイスティック2本)
東京都　小平覚

このプログラムはX-BASIC用の2人で対戦して遊ぶバトルタンクゲームです。

まず，2つのボタンがついているジョイスティックを2本用意してください。それ以外の連射ボタンがあってもいいですけど，1ボタンのものはダメですよ。

それからX-BASICを起動して，リスト1を入力してから，SAVEしてRUNでプログラムスタート。

画面設定でしばらく待ったあとに，画面上に四角い障壁の散らばっているフィールドと画面左側にプレイヤー1の赤タンク，右側にプレイヤー2の青タンクが現れます。それぞれこのタンクを操って敵を撃って生き残ったほうが勝ちです。……と，これで終わっちゃうとただの対戦タンク。

ところがこのゲーム，ちょっと攻撃方法が違うんですね。このバトルタンクではただ弾を撃つだけではなにも壊れない。敵にも当たらないのです。

攻撃をするには，Bボタンでタンクのすぐ前方に赤くて丸い泡を出します。この泡の上をタンクは通ることができません。で，Aボタンのほうを押すとタンクが白い弾を撃ちます。この弾が泡に当たると泡が大爆発を起こして，周りにあるもの，壁もタンクもみんな破壊します。この爆発にタンクが巻き込まれたら，その人の負けです。自分で起こした爆発にも巻き込まれますから，気をつけてくださいね。

むむむむむ。こいつはなかなかよくできているぞ。両方のボタンを使った合わせ技攻撃で爆発ってアイデアも秀逸だし。これだけ短いプログラムでちゃんと内容のあるゲームになってるし。自機が回転までしちゃうもんね。ちょっとスタートまでに時間はかかっちゃうけど，自機はちゃんと慣性がかかっていて，弾を撃つとあとずさりもするし。これだけよくできていると，1人プレイモードがあってもよかったような気もしますね。あと20～30行くらい増えてもいいから。敵はX，Y座標を調べて自分のタンクに寄ってきて，近づいたら泡を撃って，離れて弾を撃って爆発させる，ってルーチンだけでも十分楽しめそうな気がするし。

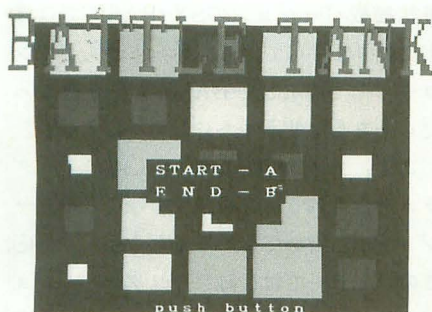
それにしても，これだけのプログラムがこの短さっていうのはすごいですよね(編集部注：リストの見やすさのため行数を少し増やしました)。ゲームスタート前にデータ作りにちょっと時間がかかるのがもどかしいけど，短くて楽しいってのはショートプロの基本ですよね。

次回作も楽しみに待ってますよん。

TWELVEでリーチなのだ！

2本目のプログラムはこれまたX-BASIC用のゲームですね。ちょっと長いけど，がんばって入力してください，TWELVE.BASです。どうぞっ！

TWELVE.BAS for X680x0
(X-BASIC(コンパイル推奨))
鹿児島県　大上幸宏

そういえば以前にこのショートプロのコーナーでMZ版のREACHってゲームを載せたことがあるような気がします。そうなんです，このTWELVE.BASはこのREACHにそっくりなルールのゲームなのです……

B-TANK.BAS

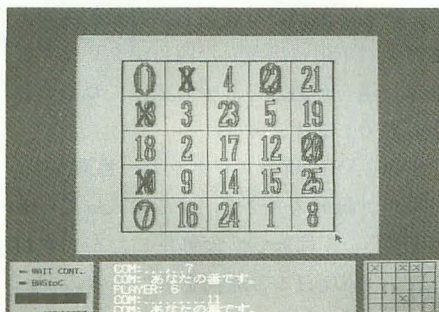
TWELVE.BAS

え？ よく憶えていたなって？ いえ，投稿原稿にそう書いてあったんですけどね(笑)。ま，移植版ってことでも別に問題なし，ですけど。

このプログラムはX-BASIC用なのですが，いろいろな処理が重いので10MHzのマシンの場合はコンパイルして遊ぶことをお勧めします。うちの24MHzのRED ZONEくらいだと，そのままでもかなり快適に遊べるくらいの速さですけどね。

それから「Z-MUSICシステムver.2.0」を買われた方はEFFECTSの効果音ファイルのなかから，BOMB.PCM，SHOT4.PCM，BOMB1.PCM，GRSCRSH.PCM，WATER.PCM, EXPL.PCM, NYAO.PCMというファイルをカレントディレクトリにコピーしておくと，より一層リッチに遊べます。なくてももちろんOKですけどね。

さて，REACHを知らない人のためにちょっとルールを説明しましょう。このゲームでは，まず2人が5×5マスにランダムに1から25までの数字を書き込みます。それから，順番に自分が言った数字にはバツを，相手が言った数字にはマルを書き込んでいきます。マルが縦横斜めいずれかに5つ(1列)揃えば勝ちなのです。そうそう，よく教室で暇なときにやる(たまには暇じゃなくてもやってチョークが飛んできたりするけど)，あれなんです。このTWELVE.BASでは対戦のお相手はX68000がしてくれますよん。

プログラムを実行させると，しばらくしてタイトル画面が表示されます。ここで初めてこのゲームをする人は，スコアデータ登録用ファイルを作成しますので，タイトル画面が表示されているときに“D”のキーを押してください。スコアのデータファイルがカレントディレクトリにSDTL.DATという名前で作られます。ゲームを始めるには，タイトル画面でSTARTにマウスカーソルを合わせ，左クリックしてください。コンパイルしてない人や10MHzのマシンだとマウスが反応しにくいので，ゆっくり踏みしめるように押すのがコツです。

で，次は5×5マスの枠のなかに，数字を入れていく作業をする画面になります。マウスを使って枠に数字を入力してください。右クリックをすると，コンピュータが自動的に数字の配列をしてくれます。数字を入力し終えると，プレイヤー，コンピュータのどちらから始めるかを聞いてくるので，マウスで選んでください。ゲームが始まったら画面の指示に従ってマウスをクリックしていけばOKです。

はぁ〜，なつかしいですねん，このゲーム。これとか，鉛筆を弾きとばして紙の上でやる大砲ゲームとかって全国共通だったんでしょうか？

さて，このプログラムですけど，前にMZ版REACHが載っていたせいもあるかもしれないですけど，とてもていねいな作りを心がけてるみたいですね。

タイトル画面でCONFIGURATIONを選ぶと，思考ルーチンの強さ，対戦回数の設定，プログラムのウェイトなどを入れられるようになっているんですけど，この設定も非常に細かくいろいろな部分が変えられるようになっているし，タイトル画面のスクロールは格好いいし，それに思考ルーチンがかなり賢いんですよね。コンピュータの思考ルーチンの強さを「セオリー」にしておくと本当に勝てないです，いや，本当に。

このプログラムは内容はよく遊ばれているゲームそのままで，オープニングや設定に凝っているんですよね。で，リストもちょっと長め(400行くらいある)。1本目のプログラムとは対象的だけど，これはこれでまたひとつのプログラムのありかたなのかな。なんか前にいったことと矛盾しているような気もするけど……。

SX-BASICの登場だっ！

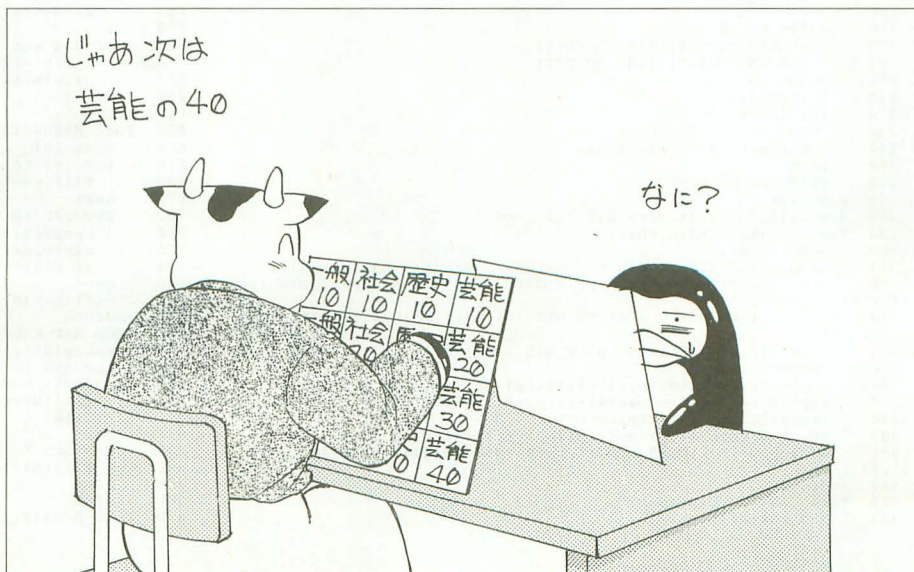
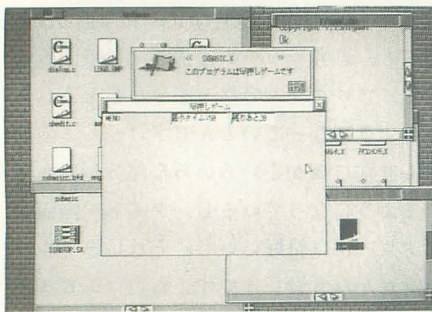
それでは今月のオーラスですね。早くも登場SX-BASIC用のプログラムです。GAME.SXBです。どうぞっ。

GAME.SXB for X680x0

(要SX-BASIC)

埼玉県　市川賢二

このプログラムは3月号の“ひなまつりPRO-68K”に掲載されたSX-BASIC(暫定版)用のプログラムです。

まず，SX-WINDOWを起動して，SX-BASICのウィンドウやシャーペンなどでリスト3を入力してください。そして，ディ

GAME.SXB

スクにSAVEしたあと，シャーペンからならファイルのアイコンをSX-BASICのウィンドウ上にドラッグしてくることでLOADして，RUNすることでプログラムがスタートします。

さてさて，ゲームがエラーなしで起動できたら遊び方を説明しましょう。このプログラムはマウスを使った早押しゲームです。[早押しゲーム]というタイトルのウインドウが現れます。このウィンドウのランダムな位置に次々と白い四角が現れますから，この四角にできるだけ早くマウスカーソルを合わせて，四角の残数が0になるまで左ボタンでクリックして消していってください。あと，最初のウィンドウの大きさのままだと右下隅に出現した四角がクリックできない場合があるので少し広げておくことをお勧めしますよ。

SX-BASIC用のプログラム一番乗り(5月号の付録ディスクで，すでに入っているけど)！　5月号で，大募集したんですが，なんとその前(この原稿を書いてるのが，5月号が本屋さんに並ぶ前だ……)に作ってしまっていたんですねー。あな恐ろしやアマグラマかな。これでSX-BASIC作者の石上氏もほっとしていることでしょう。市川さん，偉いっ。私もいま，ちょこちょことSX-BASICのプログラムを作り始めているので，ちょっとばかり悔しかったりなんかいたします，はい。

あ，そうだ。Oh!Xスタッフの協力もあって，SX-BASICのバージョンアップ版が一部のネット上にアップロードされています。パソコン通信をされている方のなかには手に入れられた方もいると思いますけど，一部のバージョンではプログラムが動かないことがたまにあります(どうも仕様じゃなくて単なるエンバグらしいです)。雑誌掲載バージョンのSX-BASICでは確実に動きますので，もしネットからダウンロードしてきたSX-BASICで動かなかったらそちらを使うようにしてくださいね。

あと，これからSX-BASIC用にプログラムを作られる方は投稿する際にどのバージョンか，書いておいてくださいね。一応，このコーナーでは付録ディスクのバージョンで載せなければいけないので……。あんまり新しいバージョンにつけられた機能を使っていたりすると，SX-BASICの新しいバージョンが付録ディスクに収録されるまで掲載を待ってもらう可能性もあります(もちろん，よくできたプログラムなら待ったあとで掲載しますけどね)。では，お待ちしてますよん。私はどんな言語のプログラムの挑戦でも受けるっ！

あー，それにしても結局ショートプロってなんなんだ！　短くてピリリと鋭いプログラムあり，ちょっと長くて凝り凝りのヤツあり。でもってニューカマーありで……。ま，いいや，作って，送って，書いて，載って楽しければそれでいいんだよね。ってことでまた来月。ぱーっと楽しくやりましょう！

リスト1　B-TANK.BAS

```
  10  /* BÀTTLE_TANK was copyrighted by KoDa  Spetial Thanks to
T.Maejima and My Bro.
  20  int a(19)={0,1,2,3, 4,5,6,7, 8,7,6,5, 4,3,2,1, 0,1,2,3 }
  30  int x(1),y(1),ox,oy,tx(1),ty(1),i,w
  40  char mu(1),m(1),v(1),tmu(1),t_f(1),c,z(1)
  50  char ch(255),j,win,game=0
  60  char M_MIN=10,M_MAX=16,MU_MIN=0,MU_MAX=15,V_MAX=9,V_MIN=0
  70  MUSIC_SET():m_play(5,6):SP_MAKE():m_stop(5,6)
  80  screen 1,1,1,1:console ,,0
  90  window(0,0,511,511):sp_disp(1):sp_on()
 100  palet(8,rgb(31,8,3)):palet(9,rgb(31,31,0)):palet(1,rgb(8,8
,8))
.110  while game=0:win=0:m_play(7,8):DISP_WRITES(win)
 120    if j=2 then {game=1:win=4} else {
 130      m_play(5,6,7,8):x(0)=64:y(0)=256:mu(0)=4:v(0)=3
 140      x(1)=448:y(1)=256:mu(1)=12:v(1)=3:z(0)=0:z(1)=0
 150      MAP_MAKE()
 160    }
 170    while win=0
 180      PLAYER_MOVE(0):TAMA_MOVE(0)
 190      PLAYER_MOVE(1):TAMA_MOVE(1)
 200    endwhile
 210    m_stop(5,6)
 220    for i=0 to 10
 230      for w=0 to 300:next
 240      m_play(1,3):m_play(2,4)
 250    next
 260    DISP_WRITES(win)
 270  endwhile
 280  m_stop(5,6,7,8):screen 2,0,1,1 :end
 290  func PLAYER_MOVE(p;char)
 300    j=stick(p+1)
 310    if m(p)<=M_MIN or m(p)>=M_MAX then m(p)=(M_MAX+M_MIN)/2
 320    m(p)=m(p)+(( j=6 )*( m(p)<M_MAX )-( j=4 )*( m(p)>M_MIN )
)*(V_MAX-v(p)+9)/3
 330    mu(p)=(( mu(p)+(( m(p)>=M_MAX )+( m(p)<=M_MIN )*15)*(-1)
) mod 16)
 340    v(p)=v(p)+( j=8 )*( v(p)<V_MAX )-( j=2 )*( v(p)>V_MIN )
 350    ox=x(p):oy=y(p)
 360    x(p)=((x(p)+(a(mu(p)+4)-4)*(v(p)-3)/3+512) mod 512)
 370    y(p)=((y(p)+(a(mu(p))-4)*(v(p)-3)/3+512) mod 512)
 380    apage(3):c=point(x(p),y(p))
 390    if c>=8 then {x(p)=ox:y(p)=oy:v(p)=( v(p)<3 )*6*(-1)}
 400    sp_set(p+10,x(p)+8,y(p)+8,mu(p)+256*(p+1)*( z(p)<>9 )*(-
1),3)
 410  endfunc
 420  func TAMA_MOVE(p;char)
 430    j=strig(p+1)
 440    if t_f(p)=1 then {m_play(1,2):v(p)=( v(p)>=3 )*(v(p)-3)*
(-1)}
 450    if t_f(p)=0 then {
 460      tx(p)=x(p)+(a(tmu(p)+4)-4)*4
 470      ty(p)=y(p)+(a(tmu(p))-4)*4:tmu(p)=mu(p)
 480    } else if t_f(p)<20 then {
 490      tx(p)=tx(p)+(a(tmu(p)+4)-4)*3
 500      ty(p)=ty(p)+(a(tmu(p))-4)*3
 510    }
 520    apage(3):c=point(tx(p),ty(p))
 530    if c=8 and t_f(p)>0 and t_f(p)<20 then BAKUHATS(tx(p),ty
(p))
 540    t_f(p)=(t_f(p)+( t_f(p)=0 )*( j=1 )+( t_f(p)>0 )*(-1))*(
 t_f(p)<10 )*( c<8 )
 550    if t_f(p)>0 and t_f(p)<20 then {
 560      sp_set(p+5,tx(p)+8,ty(p)+8,16+256*(p+1),3)
 570    } else {
 580      sp_set(p+5,0,0,0,0)
 590    }
 600    if j=2 and t_f(p)<>20 then {
 610      t_f(p)=20:apage(3)
 620      circle(tx(p),ty(p),12,8):paint(tx(p),ty(p),8)
 630    }
 640  endfunc
 650  func BAKUHATS(xx;int,yy;int)
 660    apage(3)
 670    for i=1 to 10
 680      fill(xx-i*4,yy-i*4,xx+i*4,yy+i*4,14-i/2):m_play(3,4)
 690    next
 700    for i=0 to 1
 710      apage(3):z(i)=point(x(i),y(i))
 720      win=win+( z(i)=9 )*(i+1)*(-1)
 730      if z(i)=9 then {sp_set(i+10,x(i)+8,y(i)+8,mu(i),3)}
 740    next
 750    fill(xx-40,yy-40,xx+40,yy+40,1)
 760  endfunc
 770  func MAP_MAKE()
 780    apage(3):fill(0,0,511,511,15):fill(40,40,471,471,1)
 790    for i=0 to 24
 800      ox=(i mod 5):oy=(i / 5):c=rnd()*60+20
 810      fill(90+ox*80-c/2,80+oy*80-c/2,90+ox*80+c/2,80+oy*80+c
/2,(c-6)/15+10)
 820    next
 830    for i=0 to 1
 840      fill(x(i)-16,y(i)-16,x(i)+16,y(i)+16,1)
 850    next
 860  endfunc
 870  func B_WAIT(j;char)
```

▶800円はつらい……。　　西田　浩(26)埼玉県

```
880     repeat:j=strig(1):until j=0
890     repeat:j=strig(1):until j<>0:m_play(1,2):return(j)
900   endfunc
910   func DISP_WRITES(win)
920     apage(0):symbol(0,20,"B A T T L E  T A N K",4,8,0,5,0)
930     apage(3):fill(168,232,328,296,1):cls:color  7:locate 22,
16:switch win
940       case 0:locate 22,15:print "S T A R T   -   A"
950       case 0:locate 22,17:print "E   N   D   -     B":break
960       case 1:print "W I N N E R   2 P":break
970       case 2:print "W I N N E R   1 P":break
980       case 3:print "D R A W   G A M E":break
990       case 4:print "E N D   E N D":break
1000      endswitch
1010      locate 22,28:print "p u s h   b u t t o n":j=B_WAIT(0)
1020      color 3:for i=0 to 31:locate 0,30:print " ":next
1030      apage(0):wipe()
1040    endfunc
1050    func SP_MAKE():float s,c,a
1060      screen 1,2,1,1:sp_init()
1070      locate 20,15:print "P L E A S E   W A I T"
1080      line(4,12,7,3,10):line(8,3,11,12,10)
1090      line(4,12,11,12,10):paint(7,7,10)
1100      fill(5,9,6,13,12):fill(10,9,9,13,12):fill(7,1,8,6,11)
1110      for i=0 to 15
1120        if i=8 then {
1130          wipe()
1140          line(4,3,7,12,10):line(8,12,11,3,10)
1150          line(4,3,11,3,10):paint(7,7,10):fill(5,6,6,2,12)
1160          fill(10,6,9,2,12):fill(7,14,8,9,11)
1170        }
1180        if i<8 then {
1190          a=i*(360#-22.5#)*pi()/180#
1200          c=cos(a):s=sin(a):KAITEN(c,s,i)
1210        } else {
1220          a=(i-8)*(360#-22.5#)*pi()/180#
1230          c=cos(a):s=sin(a):KAITEN(c,s,i)
1240        }
1250      next
1260      wipe():circle(7,7,4,15):fill(6,6,9,9,15)
1270      get(0,0,15,15,ch):sp_def(16,ch)
1280      sp_color(10,rgb(15,5,5),1):sp_color(11,rgb(26,26,31),1)
1290      sp_color(12,rgb(31,10,10),1):sp_color(10,rgb(5,5,15),2)
1300      sp color(11,rgb(26,26,31),2):sp_color(12,rgb(10,10,31),2
)
1310    endfunc
1320    func KAITEN(c;float,s;float,n;char)
1330      int x,y
1340      for w=0 to 255
1350        x=(w mod 16)-8:y=(w/16)-8
1360        pset(20+x+8,y+8,point(x*c-y*s+8,x*s+y*c+8))
1370      next
1380      get(20,0,35,15,ch):sp_def(n,ch)
1390    endfunc
1400    func MUSIC_SET()
1410      m_init()
1420      for i=1 to 8:m_alloc(i,1000):m_assign(i,(i+1)/2):next
1430      m_trk(1,"o0 164 v13 @51 ggag"):m_trk(2,"o0 164 v15 @14 g
<g<g>gfed")
1440      m_trk(3,"o2 116 v 8 @10 r |:255 gagb gagb gagb gagb fgfa
 fgfa fgfa fgfa:|")
1450      m_trk(4,"o1 1 8 v10 @47 r |:255 c<<c>>c16c16<<c>> c<<c<
116cccc :|")
1460    endfunc
```

リスト2 TWELVE.BAS

```
 10 /*--------------------------------------
 20 /*-      T W E L V E   L I N E S        -
 30 /*-  COPYRIGHT By Y.OUE 1994  PX9401  -
 40 /*--------------------------------------
 50 /** INIT. **
 60 screen 1,2,1,1:console 0,32,0
 70 int CX,CY,SX,SY,EX,EY,SP,EP,WC,LI,N,JT,RD,DX,DY,LB,RB
 80 int PS,P,ST,L,H,RR,YL,WF,RE,DF,JK,WD,OP,FP,T1,RG,LT
 90 char A,B,C,D,E,F1,F3,F4,F5,F7,G,GG,SC,SS,R,SF,CO,O1
 100 char FF,E1,E3,E4,K,ER,LV,FC,SN,TC,PO,AF,RF,T,F
 110 str II,IN,MM[255],P1,D1,D2,HS[255]
 120 dim char DU(25),DC(25),LX(11),PE(1,25),F2(1,11),F6(1,11)
 130 dim char F9(1,11),D3(25,1),D4(25,1),D5(11),RL(1),PP(2500)
 140 dim char BU(7000),R0(6000),R1(6500),A0(12000),A1(65000)
 150 dim str FD(2)[255],HN(4)
 160 dim str RS(5)={"","","ダブル","トリプル","",""}
 170 dim str PC(1)={"PLAYER: ","COM: "}
 180 dim str LL(11)={"12345","6789:",";<=>?","@ABCD","EFGHI","16
;@E","27<AF","38=BG","49>CH","5:?DI","59=AE","17=CI"}
 190 dim int HO(4),LS(3)={65278,49087,61423,64507}
 200 dim int PL(9)={0,21140,21140,47654,1984,4286,42280,63814,0,
65492}
 210 for I=1 to 9:palet(I,PL(I)):next
 220 palet(14,42280):palet(30,27198)
 230 HS="F.HZAZNNNY.AYAO050040030020010"
 240 error off:FP=fopen("SDTL.DAT","r"):error on
 250 if FP<>-1 then freads(HS,FP):fclose(FP):F3=1
 260 for I=0 to 4
 270   HN(I)=mid$(HS,(I*3)+1,3):HO(I)=val(mid$(HS,(I*3)+16,3))
 280 next
 290 vpage(0):apage(1):SS=val(right$(time$,2))
 300 locate 20,16:print"しばらくお待ちください"
 310 DRW():INIT():F1=1:apage(1)
 320 if (F1=1)=-1 then palet(27,2690):RG=-1 else palet(27,12042)
:RG=1
 330 vpage(3):cls:SC=val(right$(time$,2))
 340 WD=SC-SS:if sgn(WD)=-1 then WD=WD+60
 350 F1=0:if WD>8 then WD=8 else if WD=0 then F1=1
 360 G=((8-WD)*8):if G>31 then R=64-G:G=31 else R=31
 370 if R=32 then R=31
 380 if F1=0 then palet(26,rgb(R,G,0)) else palet(26,10622)
 390 PCML():pset(173,332,121)
 400 if WD=8 then WC=0 else WC=(10-WD)*2
 410 /*** MAIN ***
 420 repeat:TITLE()
 430   repeat:TC=TC+1:LOOP(500):NSET():LOOP(500):WPC():apage(0)
 440     MM=str$(TC)+"/"+str$((FC*9)+1)+"   "+str$(PO)
 450     symbol(25,461,MM,1,1,1,7,0)
 460     cls:I=1:repeat:LOOP(500):PLAYER():LOOP(500):COM()
 470     A=0:B=0:until WF=1
 480     palet(6,63814):repeat:msstat(DX,DY,LB,RB):until LB=-1
 490     INIT():palet(6,42280):if FC=0 then TC=10
 500   until TC=10:TC=0:apage(1)
 510   T1=-1:if LV=3 and FC=1 then NAMEE()
 520   for I=0 to 7:SLP("",0):next
 530   SLP("          GAME OVER",0):SLP("",0):SLP("",0)
 540   TLOOP(2):for I=0 to 4:SLP("",0):next:YL=0:PO=0
 550 until WF=1
 560 end
 570 /*** FUNC ***
 580 func PLAYER():/** PLAYER **
 590 while SF=1:SLP("COM: あなたの番です。",0)
 600   repeat:C=0:M_SC()
 610     if LB=-1 then {
 620       if PE(0,P-100)=0 then {
 630         SLP("PLAYER: "+str$(DU(P-100)),0):PCMP(0,0)
 640         PE(0,P-100)=10:PF((P-101) mod 5,(P-101)¥5,"×",4)
 650         SF=2 } else PCMP(1,0)
 660     }
 670   if P<>PS then palet(P,1984):palet(PS,42280):PS=P
 680 until SF=2:endwhile:palet(P,42280):I=1:A=0
 690 while F4=0:F4=1:repeat
 700     if DC(I)=DU(P-100) then PE(1,I)=1:A=1
 710     I=I+1:until A=1
 720   symbol(416+((I-2) mod 5)*16,416+((I-2) ¥ 5)*16,"○",1,1,1
,5,0):SFR(I-1,0)
 730 endwhile:F4=0:A=0:B=0:I=1:F5=EGP(1,0):LER(1)
 740 endfunc
 750 func COM():/** COM **
 760 if WF=0 then {  SLP("COM:",0)
 770   if LV=0 or LV=1 then RE=TCLV0()
 780   if LV=2 or LV=3 then RE=TCLV1()
 790 }
 800 I=1:while B=0:if DU(I)=DC(RE) then PE(0,I)=1:B=1
 810 I=I+1:endwhile
 820 if WF=0 then PCMP(0,0):symbol(416+((RE-1) mod 5)*16,416+((R
E-1) ¥ 5)*16,"×",1,1,1,4,0)
 830 apage(0):if WF=0 then PF((I-2) mod 5,(I-2) ¥ 5,"○",5)
 840 SF=1:if WF=0 then F5=EGP(0,0):LER(0)
 850 if WF=2 or WF=3 then WF=1
 860 endfunc
 870 func SFR(OP,O1):/** SUB.R1 **
 880 K=0:D3(OP,O1)=0:for I=1 to 25:K=K+1
 890   if D3(I,O1)<>0 then D4(K,O1)=D3(I,O1) else K=K-1
 900 next:D4(0,O1)=K
 910 endfunc
 920 func TITLE():/** TITLE **
 930 cls:SLP("START",0):SLP("CONFIGURATION",0):YL=0
 940 for I=101 to 150:palet(I,0):next
 950 mouse(1):mouse(4):msarea(0,0,511,511):setmspos(220,420)
 960 repeat:B=MSSCN(1,2,1)
 970   if B=11 then FG=0:A=1
 980   if B=12 then CONFIG()
 990 until A=1:RR=val(right$(time$,2))+(val(mid$(time$,4,2))*100
):randomize(RR)
 1000 endfunc
 1010 func NSET():/** 配置 **
 1020 cls:A=0:ST=1:setmspos(220,235):apage(0)
 1030 for I=101 to 150:palet(I,42280):next
 1040 cls:SLP("数字の配置を行ってください。",0)
 1050 SLP("(右クリック:オート)",0):YL=0
 1060 repeat:C=0:M_SC()
 1070   if RB=-1 then {
 1080     repeat:repeat
 1090       RR=int(rnd()*25)+1:until DU(RR)=0
 1100     DU(RR)=ST:ST=ST+1:until ST=26
 1110     wipe():for  I=0 to 4:for J=1 to 5
 1120     PF(J-1,I,str$(DU((I*5)+J)),3):next:next:A=1  }
 1130   if P<>PS then palet(P,1984):palet(PS,42280):PS=P
 1140     if LB=-1 then {
 1150       if DU(P-100)=0 then {
 1160         PF((P-101) mod 5,(P-101) ¥ 5,str$(ST),3)
 1170         DU(P-100)=ST:if ST=25 then A=1 else ST=ST+1
 1180       }
 1190   }
 1200 until A=1
 1210 endfunc
 1220 func WPC():/** 順番 **
```

▶ゴールデンウィークは徹夜で「アルゴス」だ！　八木　明(28)神奈川県

```
1230 palet(P,42280):A=0:ST=1:apage(1):setmspos(220,440):cls
1240 cls:SLP("先攻か後攻かを決めてください。",0)
1250 SLP("先攻",0):SLP("後攻",0):YL=0
1260 repeat:B=MSSCN(2,3,0)
1270   if B=12 then SF=1:CO=1 else if B=13 then SF=2:F4=1
1280 until SF<>0
1290 repeat:repeat
1300   RR=int(rnd()*25)+1:until DC(RR)=0
1310 DC(RR)=ST:ST=ST+1:until ST=26
1320 apage(0):for I=0 to 4:for J=1 to 5
1330   if DC((I*5)+J)<10 then C=4 else C=0
1340   symbol(400+(J*16)+C,416+(I*16),str$(DC((I*5)+J)),1,1,1,6,
0)
1350 next:next
1360 endfunc
1370 func DRW():/** DRAW **
1380 for I=0 to 63:for J=0 to 3
1390 line(0,8*I+(J*2),511,8*I+(J*2),14,LS(J))
1400 next:next
1410 fill(84,80,427,400,14)
1420 for I=0 to 4:for J=0 to 4
1430   fill(137+(J*48),117+(I*50),183+(J*48),165+(I*50),(I*5)+J+
101)
1440   box(137+(J*48),117+(I*50),183+(J*48),165+(I*50),(I*5)+J+1
26)
1450   box(138+(J*48),118+(I*50),182+(J*48),164+(I*50),(I*5)+J+1
26)
1460   palet((I*5)+J+101,42280):palet((I*5)+J+126,42280)
1470 next:next
1480 for I=1 to 4
1490   line(136+(I*48),115,136+(I*48),366,1)
1500   line(135,116+(50*I),376,116+(50*I),1)
1510 next
1520 box(136,116,376,366,1):box(135,115,377,367,1)
1530 fill(410,412,500,500,14)
1540 fill(110,412,400,500,14):fill(8,412,103,500,14)
1550 for I=0 to 5
1560   line(415+(I*16),415,415+(I*16),495,1)
1570   line(415,415+(I*16),495,415+(I*16),1)
1580 next
1590 for I=1 to 5
1600   fill(119,400+(16*I),391,416+(16*I),20+I)
1610   palet(20+I,42280)
1620 next
1630 FD(0)="01143144C0F3I0K2[0^3`1a3c1d3g3h4h0i1T0X4U1W3D1E2¥1]2
":/*  ※（注）
1640 FD(1)="70739093=1@1=3@3O1R1O3R3m1p1m3p3s1v1r3u3J3J304140414
":/* "`"は[SHIFT]+[@]
1650 FD(2)="6484:4@2H3H4I4K4L4L3R2V2`0`4d0d4g0g2i2i4p2r0r2v2v4J1
"
1660 symbol(50,487,"YUKISOFT",1,1,0,1,0)
1670 symbol(35,420,"WAIT CONT.",1,1,0,1,0)
1680 fill(20,425,26,428,26):fill(20,443,26,446,27)
1690 symbol(35,438,"BAStoC",1,1,0,1,0)
1700 for I=0 to 11:fill(221+(I*6),380,225+(I*6),384,30):next
1710 fill(15,460,95,476,1)
1720 for I=0 to 13
1730   box(LF(0,I,1)+221,LF(0,I,2)+380,LF(0,I,3)+221,LF(0,I,4)+3
80,14)
1740   line(LF(1,I,1)+221,LF(1,I,2)+380,LF(1,I,3)+221,LF(1,I,4)+
380,14)
1750   pset(LF(2,I,1)+221,LF(2,I,2)+380,14):pset(LF(2,I,3)+221,L
F(2,I,4)+380,14)
1760 next
1770 endfunc
1780 func SLP(P1;str,OP):/** PRINT **
1790 if YL=5 then {
1800   if OP=0 then { FF=0:locate 16,26:print spc(35)
1810     locate 0,31:print:locate 16,30:print P1;
1820   } else {
1830     FF=FF+1
1840     if FF<23 then print P1; else locate 16,26:print spc(35)
:locate 0,31:print:locate 16,30:print"    ";:FF=0
1850   }
1860 }
1870 if YL<>5 then {
1880   if OP=0 or OP=-1 then locate 16+OP,26+YL:print P1;:YL=YL+
1 else print P1;
1890 }
1900 endfunc
1910 func LER(OP):/** リーチ／大当たり処理 **
1920 if OP=0 then for I=126 to 150:palet(I,42280):next
1930 E1=0:if F5>0 and F5<5 then if F5>RL(OP) then E1=1:RL(OP)=F5
1940 if E1=1 then SLP(PC(OP)+RS(F5)+"リーチ"+string$(F5,"!"),0)
1950 if OP=1 then F7=1:PCMP(3,F5) else PCMP(2,F5)
1960 if OP=0 then REACH()
1970 if F5=10 and WF<>2 then SLP(PC(OP)+"当たり！！",0):A=1:B=1:
WF=3:if OP=0 then PCMP(4,0):PO=PO+10 else PCMP(5,0)
1980 if F5=10 and WF<>2 and OP=0 then { REACH()
1990   for I=0 to 11
2000     if D5(I)=2 then {
2010       SP=asc(left$(LL(I),1))-48:EP=asc(right$(LL(I),1))-48
2020       SX=160+((SP-1) mod 5)*48:SY=140+((SP-1) ¥ 5)*50
2030       EX=160+((EP-1) mod 5)*48:EY=140+((EP-1) ¥ 5)*50
2040       line(SX,SY,EX,EY,4):line(SX+1,SY+1,EX+1,EY+1,4)
2050     }
2060   next
2070 }
2080 CO=CO+1:I=1:if CO=26 and F5<>10 then SLP("引き分け",0):A=1:
B=1:WF=2:PO=PO+5
2090 endfunc
2100 func REACH():/** REACH **
2110 for I=0 to 11
2120   if D5(I)<>0 then {
2130     for J=1 to 5
2140       palet((asc(mid$(LL(I),J,1))-48)+125,1984)
2150     next
2160   }
2170 next
2180 endfunc
2190 func LF(JT,LI,N):/** FONT **
2200 RD=asc(mid$(FD(JT),(LI*4)+N,1))-48:return(RD)
2210 endfunc
2220 func MSSCN(L,H,OP):/** MENU **
2230 PS=0:repeat:mspos(CX,CY):msstat(DX,DY,LB,RB):P=point(CX,CY)
2240   if P<>PS then { for I=21 to 25:palet(I,42280):next
2250     if P>L+19 and P<H+21 then palet(P,1984)
2260   }
2270   if LB=0 and RB=0 then B=0
2280   if LB=-1 then B=10
2290   if RB=-1 then B=20
2300   if OP=1 then DEMO()
2310 PS=P:until B<>0
2320 for I=21 to 25:palet(I,42280):next:return(B+P-20)
2330 endfunc
2340 func LOOP(J):/** LOOP **
2350 for I=0 to J*WC:next
2360 endfunc
2370 func TLOOP(T):/** TIME LOOP **
2380 for LT=1 to T:SS=val(right$(time$,1)):F=0
2390   repeat
2400     if val(right$(time$,1))<>SS then F=1
2410   until F=1
2420 next
2430 endfunc
2440 func PF(X,Y,MM;str,OP):/** PRINT FONT **
2450 if len(MM)=1 then B=6 else B=0
2460 for FI=0 to 2:for FJ=0 to 2
2470   symbol(148+(48*X)+FI+B,117+(Y*50)+FJ,MM,1,2,2,2,0)
2480 next:next
2490 symbol(148+(48*X)+1+B,117+(Y*50)+1,MM,1,2,2,OP,0)
2500 endfunc
2510 func M_SC():/** MOUSE **
2520 repeat:mspos(CX,CY):msstat(DX,DY,LB,RB)
2530   apage(1):P=point(CX,CY):apage(0)
2540   if P>100 and P<126 then C=1
2550 until C=1
2560 endfunc
2570 func EGP(OP,O1):/** リーチ／当たり判定 **
2580 for I=0 to 11:LX(I)=0:next:RE=0:WF=0
2590 for I=0 to 4:for J=0 to 4:LX(J)=LX(J)+PE(OP,((J*5)+1)+I)
2600 LX(J+5)=LX(J+5)+PE(OP,(I*5)+J+1):next:next
2610 for I=0 to 4
2620   LX(10)=LX(10)+PE(OP,(I*4)+5):LX(11)=LX(11)+PE(OP,(I*6)+1)
2630 next
2640 for I=0 to 11
2650   if O1=0 then {
2660     if LX(I)=4 then F2(OP,I)=1 else F2(OP,I)=0
2670     if LX(I)=4 and OP=0 then D5(I)=1 else D5(I)=0
2680     if LX(I)=3 then F9(OP,I)=1 else F9(OP,I)=0
2690     if LV=3 and LX(I)=2 and E3=0 then F6(OP,I)=1 else F6(OP
,I)=0
2700   }
2710   if O1=1 then {
2720     if LX(I)=4 and F2(OP,I)=0 then RE=-1
2730     if LX(I)=3 and F9(OP,I)=0 then RE=-2
2740     if LV=3 and LX(I)=2 and E3=0 and F6(OP,I)=0 then RE=-3
2750   }
2760   if LX(I)=5 then WF=1:D5(I)=2
2770 next
2780 if O1=0 then for I=0 to 11:RE=RE+F2(OP,I):next
2790 if WF=1 then RE=10
2800 if LV<>3 and RE<0 then RE=-1
2810 return(RE):endfunc
2820 func TCLV0():/** 思考ルーチン　激弱／弱い **
2830 IDF():E1=0:DF=0:A=0
2840 repeat:R=int(rnd()*D4(0,1))+1:LOOP(300)
2850   if E1=0 then {
2860     if LV=0 then if F7=0 then SLP(".",1):DF=-1 else THF(10,
".!?.","..")
2870     if LV=1 then if F7=0 then THF(-1,"!.",".") else THF(10,
".!?.","..")
2880   }
2890   if E1=1 then DF=-1
2900   if DF=-1 then SLP(str$(DC(D4(R,1))),1):PE(1,D4(R,1))=10:A
=1:WF=0
2910 until A=1:ER=D4(R,1):SFR(D4(R,1),0):return(ER)
2920 endfunc
2930 func TCLV1():/** 思考ルーチン　普通／セオリー **
2940 IDF():E1=0:E2=0:C=0
2950 repeat:DF=0:R=int(rnd()*D4(0,1))+1:LOOP(300)
2960   if F7=1 then { E=0
2970     repeat
2980       R=int(rnd()*D4(0,1))+1:PE(1,D4(R,1))=1
2990       if EGP(1,1)=10 then PE(1,D4(R,1))=0:SFR(D4(R,1),1):E=
0 else PE(1,D4(R,1))=0:E=1
3000       if D4(0,1)=0 then DF=-1:E2=1:IDF():E=1:SLP("!?.",1) e
lse SLP(".",1)
3010     until E=1
3020   }
3030   if E2=0 then {
3040     if E1=0 then THF(-1,"!.",".")
3050     if LV=3 and DF=-1 and E1=0 then THF(-2,"!.",".")
3060     if LV=3 and DF=-1 and E1=0 and F7<>1 then THF(-3,"!.","
.")
3070     if E1=1 then DF=-1
```


▶まさか「ちゃだワ」に載るとは……独り言のつもりだったのに……。
吉永　紀之(21)東京都

```
3080   )
3090 until DF=-1
3100 SLP(str$(DC(D4(R,1))),1):PE(1,D4(R,1))=10:A=1:WF=0
3110 ER=D4(R,1):SFR(D4(R,1),0):return(ER)
3120 endfunc
3130 func THF(JK,D1;str,D2;str):/** SUB.R2 **
3140 PE(1,D4(R,1))=1
3150 if EGP(1,1)=JK then PE(1,D4(R,1))=0:SFR(D4(R,1),1):DF=1 els
e PE(1,D4(R,1))=0:DF=-1
3160 if D4(0,1)=0 then DF=0:E1=1:IDF():SLP(D1,1) else SLP(D2,1)
3170 endfunc
3180 func IDF():/** SUB.R3 **
3190 for I=0 to 25:D3(I,1)=D3(I,0):D4(I,1)=D4(I,0):next
3200 endfunc
3210 func DEMO():/** DEMO **
3220 for I=0 to 4:for J=1 to 5
3230 if point(J+214+ST,I+380)=30 then palet((I*5)+J+100,1984) el
se palet((I*5)+J+100,0)
3240 next:next
3250 if ST<>80 then ST=ST+1 else {
3260   ST=0:cls:palet(14,31710)
3270   for I=21 to 25:palet(I,31710):next
3280   locate 26,8:print"BEST PLAYERS"
3290   locate 24,10:print"PLAYER      SCORE"
3300   for I=0 to 4
3310     locate 25,(I*2)+12:print HN(I)
3320     locate 36,(I*2)+12:print HO(I)
3330   next
3340   TLOOP(4):cls:palet(14,42280)
3350   for I=21 to 25:palet(I,42280):next
3360   SLP("START",0):SLP("CONFIGURATION",0)
3370   YL=0:setmspos(220,420)
3380 }
3390 IN=inkey$(0):if IN="D" or IN="d" then DATA()
3400 if asc(IN)=9 then if RF=0 then RF=1:palet(6,63814) else RF=
0:palet(6,42280)
3410 if asc(IN)=27 then end
3420 LOOP(200)
3430 endfunc
3440 func CONFIG():/** CONFIGURATION **
3450 SLP("難易度     激弱 弱い 普通 セオリー",-1)
3460 SLP("対戦回数     1 10",-1)
3470 SLP("ウエイト",-1):SLP("サウンド  ON  OFF",-1)
3480 SLP("設定終了",-1):YL=0
3490 F1=0:repeat:apage(0):wipe()
3500 symbol(192+(LV*40),416,">",1,1,1,30,0)
3510 symbol(192+(FC*40),432,">",1,1,1,30,0)
3520 symbol(192+(SN*40),464,">",1,1,1,30,0)
3530 symbol(210,448,str$(WC),1,1,1,30,0)
3540 apage(1):RE=MSSCN(1,5,0)
3550 if RE=11 then if LV=3 then LV=0 else LV=LV+1
3560 if RE=12 then if FC=0 then FC=1 else FC=0
3570 if RE=14 then if SN=1 then SN=0 else SN=1
3580 if RE=13 then locate 31,28:input"WAIT";WC:locate 31,28:prin
t spc(10)
3590 if RE>20 or RE=15 then F1=1
3600 until F1=1:apage(0):wipe():cls:F1=0:B=0:A=0
3610 SLP("START",0):SLP("CONFIGURATION",0):YL=0:apage(1)
3620 endfunc
3630 func INIT():/** INIT. **
3640 for I=0 to 25:DU(I)=0:DC(I)=0:PE(0,I)=0:PE(1,I)=0:next
3650 for I=1 to 25:D3(I,0)=I:D4(I,0)=I:D3(I,1)=I:D4(I,1)=I:next
3660 D4(0,0)=25:D4(0,1)=25
3670 for I=0 to 11:for J=0 to 1
3680   F2(J,I)=0:F6(J,I)=0:F9(J,I)=0
3690 next:next
3700 CO=0:YL=0:SF=0:F5=0:F7=0:WF=0:ST=0
3710 RL(0)=0:RL(1)=0:E1=0:E2=0:E3=0:DF=0
3720 apage(0):wipe()
3730 endfunc
3740 func NAMEE():/** NAME ENT. **
3750 for I=0 to 4:if PO>HO(4-I) then T1=4-I:next
3760 if T1<>-1 then {
3770   cls:SLP("スコアが5位以内に入りました。",0)
3780   SLP("名前を入力して下さい。(3文字)",0)
3790   for I=0 to 2:IN=inkey$:locate 20+I,29:print IN:II=II+IN:n
ext
3800   if T1=0 then HO(4)=HO(3):HO(3)=HO(2):HO(2)=HO(1):HO(1)=HO
(0):HO(0)=PO:HN(4)=HN(3):HN(3)=HN(2):HN(2)=HN(1):HN(1)=HN(0):HN(
0)=II
3810   if T1=1 then HO(4)=HO(3):HO(3)=HO(2):HO(2)=HO(1):HO(1)=PO
:HN(4)=HN(3):HN(3)=HN(2):HN(2)=HN(1):HN(1)=II
3820   if T1=2 then HO(4)=HO(3):HO(3)=HO(2):HO(2)=PO:HN(4)=HN(3)
:HN(3)=HN(2):HN(2)=II
3830   if T1=3 then HO(4)=HO(3):HO(3)=PO:HN(4)=HN(3):HN(3)=II
3840   if T1=4 then HO(4)=PO:HN(4)=II
3850   HS="":for I=0 to 4:HS=HS+HN(I):next
3860   for I=0 to 4:if HO(I)=100 then HS=HS+str$(HO(I)) else HS=
HS+"0"+str$(HO(I)):next
3870   if F3=1 then FP=fopen("SDTL.DAT","c"):fwrites(HS,FP):fclo
se(FP)
3880 }
3890 endfunc
3900 func DATA():/** SCORE DATA **
3910 FP=fopen("SDTL.DAT","c"):fwrites(HS,FP):fclose(FP)
3920 repeat:until inkey$(0)=""
3930 endfunc
3940 func PCML():/** PCM DATA **
3950 error off:FP=fopen("WATER.PCM","r"):error on
3960 if FP=-1 then AF=0:SN=1 else AF=1:fread(PP,2501,FP):fclose(
FP)
3970 if AF=1 then {
3980   FP=fopen("NYAO.PCM","r"):fread(BU,7001,FP):fclose(FP)
3990   FP=fopen("SHOT4.PCM","r"):fread(R0,6001,FP):fclose(FP)
4000   FP=fopen("EXPL1.PCM","r"):fread(R1,6501,FP):fclose(FP)
4010   FP=fopen("GRSCRSH.PCM","r"):fread(A0,12001,FP):fclose(FP)
4020   FP=fopen("BOMB1.PCM","r"):fread(A1,65001,FP):fclose(FP) }
4030 endfunc
4040 func PCMP(GG,OP):/** PCM PLAY **
4050 if SN=0 then {
4060   if GG=0 then a_play(PP,4,3)
4070   if GG=1 then a_play(BU,4,3)
4080   if GG=2 then a_play(R0,4,3)
4090   if GG=3 then a_play(R1,4,3)
4100   if GG=4 then a_play(A0,4,3)
4110   if GG=5 then a_play(A1,4,3)
4120 }
4130 endfunc
```

リスト3 GAME.SXB

```
 1: ▼Window Size (300,200),1,0,早押しゲーム
 2: int hiscore=50
 3: int count=0
 4: int max=20
 5: int stime
 6: stime=gettime()
 7: Text2.caption="残りあと"+str$(max-count)
 8: Text3.caption="最小タイム:"+str$(hiscore)
 9: /*  ここで、初期化に必要な処理を行なって下さい
10: func File_Drop(filename;str)
11: endfunc
12:
13: ▼1,Text1 (0,0,32,14),0,0,3,0,0,1, MENU
14: func Text1_Click()
15: Text4.move=0,15,104,31
16: Text5.move=0,32,104,47
17: endfunc
18:
19: ▼1,Text2 (228,0,300,16),0,0,3,0,0,1,
20: ▼1,Text3 (120,0,216,16),0,0,3,0,0,1,
21: ▼1,Text4 (400,0,504,16),0,0,3,0,0,1,もう一回最初から
22: func Text4_Click()
23: count=0
24: stime=gettime()
25: Text4.move=400,0,504,16
26: Text5.move=400,17,504,33
27: Text2.caption="残りあと"+str$(max-count)
28: endfunc
29: ▼1,Text5 (400,17,504,33),0,0,3,0,0,1,このプログラムは
30: func Text5_Click()
31: Text4.move=400,0,504,16
32: Text5.move=400,17,504,33
33: alart(1,"このプログラムは早押しゲームです")
34: exit()
35: endfunc
36: ▼1,Text6 (150,100,155,105),0,0,3,0,0,0,
37: func Text6_Click()
38: int x
39: int y
40: x=rnd()*295
41: y=rnd()*175+20
42: Text6.move=x,y,x+5,y+5
43: count=count+1
44: Text2.caption="残りあと"+str$(max-count)
45: beep
46: if count=max then gameover()
47: endfunc
48:
49: func gameover()
50: int etime
51: int score
52: int flg
53: str mes
54: etime=gettime()
55: score=etime-stime
56: if score<hiscore then hiscore=score:mes="HighScore!":flg=&h
101 else mes="GameOver!!":flg=1
57: mes=mes+chr$(13)+"Score:"+str$(score)
58: alart(flg,mes)
59: Text3.caption="最小タイム:"+str$(hiscore)
60: count=0
61: stime=gettime()
62: endfunc
63:
64: func gettime()
65: str t,h,m,s
66: t=time$
67: h=left$(time$,2)
68: m=mid$(time$,4,2)
69: s=right$(time$,2)
70: return(val(h)*3600+val(m)*60+val(s))
71: endfunc
72:
```

▶今年中(できれば8月くらいまで)にNew X68000は出ないんでしょうかね～。資金はなんとか貯まったのに……。
菅野 誠(21)大阪府

SX-BASIC公開デバッグ第4回

開発キットとこいのぼりPRO-68K

Ishigami Tatsuya 石上 達也

機能拡張されたSX-BASIC。ここでは新しく加わった関数や基本機能をまとめて見ていきます。そのほか，ウィンドウデザイナが出力するアイテム指定行の詳細も解説します。

ソースリスト

読者の皆さま，関係者各位のおかげをもちまして，3月号よりも大幅にデバッグの進んだバージョンを5月号に掲載させていただくことができました。

ただ，公開デバッグといいながら，先月号ではディスクの容量の関係上，ソースプログラムを収録することができませんでした。

ソースリストが必要で，パソコン通信に加入されている方はNetwork-SX，あるいは，PC-VANのX1clubに最新バージョンのソースリストをアップさせていただきましたので，そちらをご利用ください。

こいのぼりPRO-68K

「こいのぼりPRO-68K」に収録されたバージョンには，若干の拡張・変更が施されています。

●追加された関数

findtskn(文字列，式)

戻り値：int

文字列と同じファイル名を持つタスクを探します。検索は式で示されるタスクIDから始め，タスクIDの大きなほうへ行います。起動されているプログラムすべてを対象に検索を行う場合には，式に0を指定してください。見つかった場合にはそのタスクのIDを，見つからなかった場合には−1を返します。タスクの検索にはワイルドカードが使えます。

例)

```
/* ワイルドカードに合致するプログラムの数を数える
int func CntTskn(name;str)
  int id = 0,cnt = 0
  while(1)
    id = findtskn(name, id)
    if(id == -1) then break
    cnt = cnt + 1
  endwhile
  return(cnt)
endfunc
```

getenv(文字列，式)

戻り値：str

式で指定した環境の文字列に指定した環境変数を得ます。式に0を指定することにより自分の環境を指定することができます。

例)

```
print "テンポラリバッファは"; getenv("temp",0);"です"
```

setenv(文字列1，式，文字列2)

戻り値：int

文字列1で指定した環境変数に文字列2を設定します。文字列2に" "(空文字列)を指定すると環境変数を消去します。文字列2の長さは最大255文字までです。

式の値により環境ポインタを変更することができます。自分の環境に設定する場合は式に0を指定してください。

例)

```
setenv("temp", 0, "c:")
```

●追加されたシステム変数

shiftkeybit

現在のキーボードの状態を表すシフトキービットを返します。シフトキービットの内容は図1のとおりです。ロック系(LEDのあるもの)のキーは，LEDが点燈で1，消燈で0となります。その他のキーは，押されている間だけ1になります。

例)

```
/* shiftキーが押されているか調べる
func isShift()
  return(shiftkeybit and 1)
endfunc
```

●変更されたシステム変数

taskid

「ひなまつりPRO-68K」に収録されたバージョンではウィンドウエンジンのタスクIDを返していましたが，SX-BASIC本体のタスクIDを返すようになりました。

●追加されたメソッド

active

テキストアイテムをアクティブ状態(ウィンドウエンジンがアクティブならカーソル点滅状態)にします。

このメソッドを持つのはテキストアイテムだけです。

例) Text1.active

アイテム指定行の内容

SX-BASICのプログラムには，「▼」で始まり行の終わりと共に終わるX-BASICでは見かけなかったような文字列が埋め込まれています。

図1 シフトキービット

b31〜b21	b20	b19	b18	b17	b16
0	XF5	XF4	XF3	XF2	XF1

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	全角	ひら	INS	CAPS	コード	ローマ	かな	CAPS	コード	ローマ	かな	OPT2	OPT1	CTRL	SHIFT

これはウィンドウエンジンに渡すアイテムの配置に関する情報です。SX-BASIC実行時には，プログラムの実行に先だってウィンドウエンジンに渡されます。

本来なら，これらの情報は常にウィンドウデザイナに読み込まれ編集されるべきなのですが，SX-BASICからちょっとした変更を加えたいというときもあります。また，SX-BASICの支援ツールなどを作成する際，これらの意味がわかると便利な場合があるかもしれません。そのようなときには以下の内容を参考にしてください（引数名の後ろにstrとあるのはstr型文字列定数で，それ以外はint型定数として扱われます）。

▼Window Size (X1,Y1),growbox,drag,caption

SX-BASIC用のファイルは必ずこの行から始まります。この行を認識することにより，ウィンドウデザイナやウィンドウエンジンは自分のウィンドウをオープンします。

(X1,Y1):ウィンドウの大きさ。X1がウィンドウの横幅，Y1がウィンドウの高さを表します。

growbox：ウィンドウサイズ可変フラグ。この値が0のとき，ウィンドウの大きさは固定されます。それ以外ではウィンドウ右下のサイズボックス（画面上には表示されませんが，内部的には存在します）により大きさを変更できるようになります。この場合，サイズボックス（のあるところ）をダブルクリックすることにより，画面の大きさいっぱいにウィンドウを拡大することができます。その状態でダブルクリックすると元の大きさに戻ります。

drag：この値が0のとき，ファイルアイコンのドロップを禁止します。それ以外の場合にはドロップを許可します。SX-BASIC実行時，ファイルアイコンがドロップされるとFileDrop(file_name;str)関数が実行されます。

caption(str)：ウィンドウのタイトル。

▼1,Name,(X1,Y1,X2,Y2),Isarray,Index,fontsize,backcolor,forecolor,editable,arrange,frame,caption

1：以下に続くデータがテキストアイテム（テキスト表示窓）用のものであることを示します。変更してはいけません。

Name(str)：アイテムの名前。

(X1,Y1,X2,Y2)：アイテムの座標。図2で示された領域にアイテムが配置されます。

Isarray：このアイテムが配列化されているか否かを示すフラグ。0で非配列アイテム，それ以外で配列アイテムです。

Index：配列アイテムの配列番号。非配列アイテムの場合，意味はありません。

fontsize：フォントの種類。

0＝12ドット
1＝16ドット
2＝24ドット

となっています。

backcolor：背景色。

0＝白
1＝薄いグレー
2＝濃いグレー
3＝黒
4＝黄色
5＝赤
6＝緑
7＝青

となっています。上記以外の値を指定しないでください。

forecolor：文字色（値はbackcolorと同じ）。

editable：編集可フラグ。0でキーボードによる編集不可，それ以外で可能となります。

arrange：行揃えモードの指定。

0＝左寄せ
1＝中央寄せ
－1＝右寄せ

となっています。上記以外の値を指定しないでください。

frame：枠を描くかどうかのフラグ。0で枠なし，それ以外で枠を描画するようになっています。

caption(str)：キャプションに指定した文字列。

▼2,Name,(X1,Y1,X2,Y2),Isarray,Index, height

2：以下に続くデータがレクタングルアイテム用のものであることを示します。変更してはいけません。

Name(str)：アイテムの名前。

(X1,Y1,X2,Y2)：アイテムの座標。図2で示された領域にアイテムが配置されます。

Isarray：このアイテムが配列化されているか否かを示すフラグ。0で非配列アイテム，それ以外で配列アイテムです。

Index：配列アイテムの配列番号。非配列アイテムの場合，意味はありません。

height：「彫り」の高さ。負の値で彫り込み，正の値で飛び出します。範囲は－5～5です。

▼3,Name,(X1,Y1,X2,Y2),Isarray,Index,caption

3：以下に続くデータが標準ボタンアイテム（文字ボタン）用のものであることを示します。変更してはいけません。

Name(str)：アイテムの名前。

(X1,Y1,X2,Y2)：アイテムの座標。図2で示された領域にアイテムが配置されます。

Isarray：このアイテムが配列化されているか否かを示すフラグ。0で非配列アイテム，それ以外で配列アイテムです。

Index：配列アイテムの配列番号。非配列アイテムの場合，意味はありません。

caption(str)：キャプションに指定した文字列。

▼4,Name,(X1,Y1,X2,Y2),Isarray,Index,min,max,value

4：以下に続くデータがボリュームアイテム（スライドボリューム）用のものであることを示します。変更してはいけません。

Name(str)：アイテムの名前。

(X1,Y1,X2,Y2)：アイテムの座標。図2で示された領域にアイテムが配置されます。

Isarray：このアイテムが配列化されているか否かを示すフラグ。0で非配列アイテム，それ以外で配列アイテムです。

Index：配列アイテムの配列番号。非配列アイテムの場合，意味はありません。

min：最小値の指定。valueやmaxより大きくてはいけません。

max：最大値の指定。valueやminより小さくてはいけません。

value：初期時の値の指定。

min≦value≦max

を満たしていなければなりません。

▼5,Name,(X1,Y1,X2,Y2),Isarray,Index,value

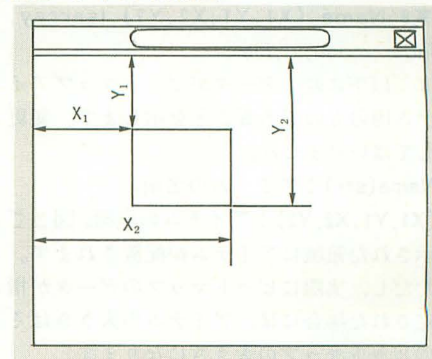
5：以下に続くデータがセレクトボタンアイテム（チェックボックス）用のものであることを示します。変更してはいけません。

Name(str)：アイテムの名前。

(X1,Y1,X2,Y2)：アイテムの座標。図2で示された領域にアイテムが配置されます。

Isarray：このアイテムが配列化されているか否かを示すフラグ。0で非配列アイテム，それ以外で配列アイテムです。

Index：配列アイテムの配列番号。非配列ア

図2　アイテムの配置

イテムの場合，意味はありません。
value：初期時の状態。0で「解除」状態，それ以外で「設定」状態を表します。
▼6,Name,(X1,Y1,X2,Y2),Isarray,Index,value
6：以下に続くデータがオルタネイトボタンアイテム(ラジオボタン)用のものであることを示します。変更してはいけません。
Name(str)：アイテムの名前。
(X1,Y1,X2,Y2)：アイテムの座標。図2で示された領域にアイテムが配置されます。
Isarray：このアイテムが配列化されているか否かを示すフラグ。0で非配列アイテム，それ以外で配列アイテムです。
Index：配列アイテムの配列番号。非配列アイテムの場合，意味はありません。
value：初期時の状態。0で「解除」状態，それ以外で「ON」状態を表します。
▼7,Name,(X1,Y1,X2,Y2),Isarray,Index,min, *max*,*value*,editable,caption
7：以下に続くデータがアップダウンボタンアイテム(数値調整ボタン)用のものであることを示します。変更してはいけません。
Name(str)：アイテムの名前。
(X1,Y1,X2,Y2)：アイテムの座標。図2で示された領域にアイテムが配置されます。
Isarray：このアイテムが配列化されているか否かを示すフラグ。0で非配列アイテム，それ以外で配列アイテムです。
Index：配列アイテムの配列番号。非配列アイテムの場合，意味はありません。
min：最小値の指定。valueやmaxより大きくてはいけません。
max：最大値の指定。valueやminより小さくてはいけません。
value：初期時の値の指定。

min≦value≦max

を満たしていなければなりません。
editable：編集可フラグ。0でキーボードによる編集不可，それ以外で可能となります。
caption(str)：キャプションに指定した文字列。
▼8,Name,(X1,Y1,X2,Y2),Isarray,Index,mode,id,file
8：以下に続くデータがビットマップアイテム用のものであることを示します。変更してはいけません。
Name(str)：アイテムの名前。
(X1,Y1,X2,Y2)：アイテムの座標。図2で示された領域にアイテムが配置されます。ただし，実際にビットマップのデータが指定された場合には，アイテムの大きさはそのビットマップの大きさになります。
Isarray：このアイテムが配列化されているか否かを示すフラグ。0で非配列アイテム，それ以外で配列アイテムです。
Index：配列アイテムの配列番号。非配列アイテムの場合，意味はありません。
mode：描画モードの指定。

- 0：標準
- 1：反転
- 2：強調
- 3：強調反転
- 4：消去
- 5：マスク無視
- 6：網掛け
- 7：網掛け反転
- 8：不可視
- 9：不可視反転

となっています。上記以外の値を指定しないでください。
id：ビットマップデータを収めたリソースIDの指定。リソースを使用しない場合は0にします。
file(str)：ビットマップデータを収めたファイルの指定。ファイルから読み込まない場合は空文字列にしておきます。
▼9,Name,(X1,Y1,X2,Y2),Isarray,Index,id
9：以下に続くデータがリストボックスアイテム用のものであることを示します。変更してはいけません。
Name(str)：アイテムの名前。
(X1,Y1,X2,Y2)：アイテムの座標。図2で示された領域にアイテムが配置されます。
Isarray：このアイテムが配列化されているか否かを示すフラグ。0で非配列アイテム，それ以外で配列アイテムです。
Index：配列アイテムの配列番号。非配列アイテムの場合，意味はありません。
id：メニューデータを収めたリソースIDの指定。リソースを使用しない場合は0にします。
▼10,Name,(X1,Y1,X2,Y2),Isarray,Index,interval
10：以下に続くデータがクロックアイテム用のものであることを示します。変更してはいけません。
Name(str)：アイテムの名前。
(X1,Y1,X2,Y2)：アイテムの座標。図2で示された領域にアイテムが配置されます。SX-BASIC実行時には，クロックアイテムは表示されませんので意味はありません。
Isarray：このアイテムが配列化されているか否かを示すフラグ。0で非配列アイテム，それ以外で配列アイテムです。
Index：配列アイテムの配列番号。非配列アイテムの場合，意味はありません。
interval：クロックアイテムのインターバルタイムの指定。単位は1/100秒です。

ウィンドウエンジンとの通信

「▼」で始まるデータは，SX-BASICが実行時にウィンドウエンジンに送るデータです。つまり，SX-BASICでなくてもほかのアプリケーションからでもウィンドウエンジンに対して同じようなデータをタスク間通信で送ってやればウィンドウエンジンの機能を利用できるわけです。ウィンドウエンジンに対して，マウスからのクリック，キーボードの入力などが行われた際には，

"@DIRECT"＋
アイテム名_イベント名(各種の引数)

といった文字列が親タスクに対して送られてきます(4月号参照)。

SX-BASICは，それをダイレクトコマンドとして処理しているわけですが，これらの文字列を分解できれば必要なデータは取り出せるわけです(C言語ならsscanf()関数でできますね)。

SX-BASIC本体とウィンドウエンジンの連携はこれだけです。ウィンドウデザイナとSX-BASIC本体との連携は，普通のSX-BASICプログラムと同じ内容のものがテンポラリファイルを通じて渡されるだけです。特に変わったことはしていません。

開発キットツール集

今春，ようやく「開発キット」がリリースされ，あわせて「開発キットツール集」も発売されました。内容は，先月号の田村健人氏による紹介記事を見ていただくとして，SX-BASICとの関係を考えてみます。特に，リソースの扱いが容易になったことはSX-BASICの環境にも反映させるべきだと思います(ただし，皆さんの出費も増えてしまいますが……)。

リソースの扱い

システムの構造などから見て，SX-WINDOWは，Macintoshのさまざまなアイデアを模倣しています。

開発キットのリソースエディタ.XとMacintoshのResEditの類似性から見ても，リソースタイプの種類からいっても，SX-WINDOWのリソースの概念はMacintoshからの借りものであることに間違いはなさそうです。

で，そのリソースに関して，

<A>

たとえば，ババヌキのゲームを行っている横で，七並べをやっている場合を考えてください。トランプのビットマップデータは，どちらのタスクでも同じはずなのに，それぞれのタスクが独自に持っていたのでは，メモリ空間内に同じデータが2つも存在し，不経済です。データを共有できれば，メモリを有効に活用することができそうです。さて，複数のタスクで同じデータを共有する場合，それらのタスク間でデータの取扱方法を同じにしておかなければなりません。その方法にある程度の一般性を持たせることができれば，さらに多くのプログラムでデータを共有できるかもしれません……。

<B>

たとえば，あなたの使っているアプリケーションで，気にいらない表示があったとします。プリンタの設定を行うメニュー項目がアプリケーションによっては「プリンタ設定」だったり「用紙設定」だったり「ページ設定」だったり「Page SetUp」だったりします。このようなささいなところを変更する場合を考えてください。普通なら巨大なプログラムのなかから，該当する文字列を見つけ出し，変更を加えて再コンパイルを行わなければなりません。これはソースコードが手元にあった場合の話で，市販ソフトのような場合は，まず逆アセンブラなどを用いた解析作業を行わなければなりません。ところが，このメニュー項目のデータがプログラムとは別に管理されていれば……。

といった説明がなされていますが，どうやら，リソース導入のいちばんの切実な原因は，

ThinkPASCALは構造体の初期化付き宣言ができなかった

ことの回避手段のようです。ThinkPASCALとは，Macintoshのアプリケーション作成の公式言語で，Macintoshの技術資料もほんの数年前まではPascalを用いて説明がなされていたそうです。

さて，SX-BASICでは構造体の初期化どころか，構造体という考え方すらサポートしていません。SX-WINDOWで扱うデータは構造化されていないもののほうが珍しいくらいですから，SX-WINDOWの機能をフルに使ったプログラムを作成するのは困難です。

たとえば，SX-WINDOWではコントロールを扱うのに，親ウィンドウのウィンドウポインタ，領域を表すレクタングルデータ，可視化フラグ，ハイライトフラグ，コントロールの持つ値，最小値，最大値……，といろいろな種類のデータを使用します。

SX-BASICでは，コントロールの制御を行おうとする場合，これらのデータに相当するプロパティの設定を行います。ウィンドウデザイナを使わずに同じことをやろうとする場合，

```
Vol1.new = 3,10,10,50,50
Vol1.max = 10
Vol1.min = 0
Vol1.value = 5
Vol1.visible = 1
  ⋮
```

と，しなければなりませんでした。

ところが，あらかじめこれらのデータをリソースとして用意しておけば，「何番のリソースと同じデータ」と指定するだけで同じ効果が得られるようになるわけです。

この場合，リソースファイルとは，

ID=128：座標が(20,20,60,40)で，初期値が1のチェックボタン。

ID=129：座標が(40,60,60,80)で，キャプションが「確認」の文字ボタン。

⋮

のような対応表です。つまり，あらかじめ予想されるプロパティ設定の組み合わせをすべてリソース登録しておけば（若干のプロパティ変更なら可読性を落とさずに行えるので，厳密にすべての組み合わせではありません)，SX-WINDOW固有のデータについては構造体がなくても扱えるようになるわけです。

コードリソース

リソースとは，多くの場合なんらかのデータを扱います。たとえば，ウィンドウデザイナのボタンは，WIND.LBというファイルにリソースデータとして入っています。つまり，リソースエディタなどを持っていれば，自由に書き換えることができるのです。

で，考えてみれば，パソコンというのはしょせん「0」と「1」の世界ですから，プログラムコードも，データも変わりはありません。というわけで，ビットマップデ

BEDIT.X

BEDIT.Xとは，ウィンドウデザイナからコード入力を行うテキストエディタです。SX-WINDOWにもいろいろなテキストエディタがあります。私がウィンドウデザイナ用に作ったテキストエディタでは満足できない人も多いでしょう。そこで，ウィンドウデザイナにテキストエディタを組み込まずに，別のプログラムとしました。ですから，自作のテキストエディタなどを愛用されている方は以下に述べる機能を追加し，BEDIT.Xとリネームするだけでそのエディタを用いてコードを入力することができるようになります。

また，市販のエディタなどで改造できないものも，間にパラメータを変更するプログラム（以下，インタフェイスプログラム）をはさんでやれば使用することができます（不可能なエディタも考えられます)。

・起動

BEDIT.X -Hnn

nnは編集すべきテキストが収められたハンドルです。ハンドルの作成・破棄はウィンドウデザイナが行うのでテキストエディタが行ってはいけません。どうしても行ってしまう場合には，インタフェイスプログラムで新たにハンドルを作成し，データをコピーしてください。エディタには新しいハンドルを渡すことによって，この問題を回避できます。また，ファイルデータしか扱えないエディタを使用する場合にも，インタフェイスプログラムで渡されたデータをいったんファイルに書き出してください。

編集すべきテキストは，すべてこのようにエディタに渡されます。ウィンドウデザイナからすでに起動しているエディタにテキストが渡されることはありません。テキストエディタ間でうまく連絡がとれれば，コード全域を範囲とする検索・置換が行えます（BEDIT.Xはそこまではやっていません)。

・タスク間通信

ウィンドウデザイナはSX-BASIC方式のタスク間通信によってエディタに命令を伝えます。伝えられる命令の種類は以下の2種類です。

ACTIVE

エディタをアクティブにしてください。

市販ソフトなどで，このような改造を行うことができない場合には，インタフェイスプログラムからエディタにアクティベートイベントを送りつけることで実現できると思います。実現できなければこの命令は無視してもかまいません。

SAVE

テキストの保存を行ってください。保存は起動時に指定されたハンドルへ行います。このとき，編集されたテキストが与えられたハンドルのサイズよりも大きい場合にはサイズを変更しても構いません。

市販ソフトなどで，このような改造を行うことができない場合には，インタフェイスプログラムでエディタをアクティブにし，OPT.1+「S」,「リターンキー」とキーダウンイベントを発生させることで同様の機能を実現できると思います。

・終了

特に必要なことはありません。そのままウィンドウを閉じて終了してください。

ータやメニューデータと同じように，プログラムも複数のアプリケーションで共有したりできるようになります。これが，コードリソースです。
いままで，SX-WINDOWで共有したり，書き換え可能にしたりすることによって，メリットのあるデータというのはほとんどありませんでした。あるいは，リソース化するよりは，特定のファイルにしておけば十分といったものばかりです。
しかし，コードリソースに関しては事情が変わってきます。
SX-WINDOWでコントロールやダイアログの描画を行うルーチンは「SYSTEM.LB」というファイルに収められています。また，プリンタドライバもここに収められています。
たとえば，プリンタを買い換えたので，プリンタドライバも変えなければならない，といった状況を考えてみてください。もし，シャーペンやEasydrawなどのアプリケーションにプリンタ制御部分が書き込まれていた場合，それらを探し出して片っ端から書き換えなくてはなりません。
しかし，現実にはそういったものはコードリソースとして1カ所にまとめられているので，その1カ所を書き換える（というか，別なものに取り換える）だけで済むようになっています。
また，SX-WINDOWのコントロールが気にいらなければコントロール関連のリソースを書き換えて，やることによって好みのものに変更することも可能です。
つまり，コードリソースを作成できるようになったというのはSX-WINDOWのシステム部分を改造・拡張できるようになったということを意味します。
事実，開発キットツール集でサポートされた「フロートウィンドウ」[1]や「デラックスメニュー」[2]は，リソースとして供給されていました。SX-WINDOW本体にはいっさい手を加えずに，リソースの追加だけで機能拡張が行えたのです。
いままでは，ほとんど，

1アプリケーション＝1リソースファイル

という関係がありました。「プログラム＝アルゴリズム＋データ構造」という考えがありますが，これまでは，その「データ」部分しかリソース化することができなかったからです。つまり，形のうえからだけデータを共通化/共有しても，そのデータをどのように解釈するかという部分が共有できなかったのです。

現在，データをどのように解釈するかという部分もリソース化できるようになりましたので，今後は複数アプリケーションによる単一リソースの共有化，という話もさらに現実味を帯びてきます。DLLとかDDEなどという話も高嶺の花ではなくなるかもしれません（しかし，個人でやるにはちょっと……）。

1） サブウィンドウみたいなものだが，はるかに簡単に扱うことができる。
2） 階層化メニューのサポートやメニュー中でのグラフィックの使用などが可能となったもの。

コードリソースコンバータ

開発キットツール集には，コードリソースコンバータというツールが収録されていました。これは，Cコンパイラなどで出力される普通の＊.X型データを，（疑似的に）リエントラントな＊.R型データへと変換するものです。
少し専門的になってしまいましたが，要するに，

SX-BASICからマシン語が使えるようになる

ということを意味します（「こいのぼりPRO-68K」に収録されたバージョンでは扱えませんが，次回のバージョンでは，マシン語サブルーチンを扱えるようにSX-BASIC側の整備を行っておきます）。
SX-BASICは中間言語コンパイラという構成上，実行速度はあまり有利とはいえません。ただでさえSX-WINDOWは限られた処理能力をタスクの頭数で割って使用します。ひとつでものろまなタスクがあると，ほかのタスクまでが迷惑してしまいます。

ボタンが押された
↓
テキストに「ボタン1が押された」と表示
↓
終了

のような短い処理では問題ないのですが，楽譜エディタのようにたびたび考え込まれていたのでは，その都度ほかのタスクまで止まってしまいます。
楽譜エディタで処理時間のかかる場所はわかっています。音符の並べ変え部分です。この部分だけでも高速化できれば，操作性は著しく向上します。
しかし，SX-BASICは，
1） コンバータがないのでC言語へ変換できない。
2） 外部関数の作成が行えない。
ので，X-BASICのような方法は使えませんでした。
しかし，コードリソースをSX-BASICから呼び出すことができれば，SX-BASICのプログラムがマシン語並みの速度で実行できるようになるわけです。

プロトタイプ作成ツールとしてのSX-BASIC

SX-BASICからC言語へと変換するコンバータが完成すれば，市販のアプリケーションと同様のものが，BASICで作れるようになるかもしれません。しかし，SX-BASICの最終的な仕様も決定しておらず，就職活動という個人的な事情からも完成が危ぶまれています(いまのところプロトタイプはおろか，1行もプログラムしていない)。いつかは完成するような気がするのですが，いつ完成になるのかは私にもわかりません。
また，SX-BASICには「ポインタ」の概念がありませんので，コンバータから出力されるC言語のプログラムにも，ポインタは使われません。近い内容はすべて配列で処理されます。このほかにもBASICとC言語にはさまざまな相違点があり，コンバータにより自動生成されたC言語のプログラムと，ある程度の経験のあるプログラマが作成したプログラムとでは品質が違ってきます。
プログラムにそれなりの品質(大きさや実行速度など）が要求される場合には，どうしてもC言語に頼らなければならない場面も出てくるでしょう。
そのような場合，まずウィンドウのオープンなどといったあまり本質的でない部分をSX-BASICで記述し，核となる（かつ，高速性が要求される）部分をC言語で記述します。先ほどのコードリソース呼出機能を使えば，SX-BASICとC言語の連携プレーは可能となりますから，これでもプログラムは，一応動作するでしょう。その後，同じ要領で1つひとつの関数をC言語へ変換していけば，すべてC言語で作成されたプログラムができあがります。
SX-WINDOWのアプリケーションはコンパイルに時間がかかり，トライ＆エラーするには非能率的だといえます。実際に動作させなければ決められない定数は，SX-BASICで適当に調整してから，C言語へ組み込みます。実際，このような定数は主にウィンドウデザイン関係に多いのですが，これはSX-BASICのもっとも得意な分野です。
私自身，このような方法でアプリケーションを作成したことはないので断言はでき

ませんが，おそらく初めからC言語を用いるよりは，このほうがはるかに効率が上がるものと思われます。

ミックスドランゲージの可能性

これまでは，主にプログラムの一部をC言語で作成する場合の話でしたが，コードコンバータは，＊.X型の実行ファイルならたいていのものは扱えるわけです。

たとえば，以前本誌に藤井義巳・藤木健士両氏によるPurePASCALが発表されました。このプログラムを使えば，PASCALでSX-WINDOWのプログラムが作成できるわけです。

また，GNUプロダクトのなかにf2cという，FORTRANからC言語へ変換を行うプログラムがあります。これを用いて，FORTRANによるSX-WINDOWプログラミングも可能となります。特にFORTRANには数値演算関係の蓄積が莫大にありますから（しかも，たいていは読みにくいプログラムで，ほかの言語へ変換するのは困難），新しい分野への可能性も開けるかもしれません。

来月号……

本来なら，SX-BASIC入門から始めて，やがてシステム情報の解説へと進めるべきだったのかもしれません。しかし，SX-BASICのサポートツールを作成してくださる方にはなるべく早く技術情報を伝えたかったので，結局，入門記事と解説記事が混じったターゲットのよくわからない記事が続いてしまいました。

今月号でSX-BASICのサポートツールを作成するのに必要と思われる情報はひととおり発表しました（プログラムの主な関数の紹介などが残っていますが，もう少しシステム全体の仕様が煮詰まってからということにします）。

来月号からは，X-BASICはわかるけどSX-WINDOWのプログラムは初めてという読者のための説明を中心に行っていきたいと思います。そのほか，引き続きSX-BASICのバグ報告や要望などもお待ちしていますのでよろしく。

こいのぼりPRO-68K不都合点一覧

5月号の付録ディスクに収録されたプログラムでいくつかの不都合がありましたのでまとめて紹介しておきます。

●MELODIUS.SXB

楽譜エディタのリソース対応が不十分なまま収録されていました。

以下，修正の手順を説明します。

まず，ファイルをシャーペンなどに読み込み，プログラム内から“pt4”の文字列を探して，ファイル名の直前のIDを設定してください（IDは5月号41ページ参照）。たとえば，

```
▼8,Bitmap2 (552,40,590,72),0,0,0,0,sb2.pt4
```

ならば，

```
▼8,Bitmap2 (552,40,590,72),0,0,0,3001,
```

のようになります。

次に“Bitmap53”の定義されている部分を探し，上にあるものから順に1000～1015のIDを割り当てます。

リスト1に実際の対応を示します。参考にしてください。

また，普段，環境変数“temp”を設定していない人はgetenv()関数を取り除いてください（2箇所あります）。

メモリ2Mバイトで動作させる場合はZ-MUSICの常駐量を最小にしておいてください（ZMUSIC-T20 -P0など）。どうしても足りない場合は1トラック仕様に改造したほうがいいでしょう。2次元配列で，

```
～(2,1288)
```

のようになっているところを，

```
～(1288)
```

などの変更を加え，参照部分を変更していってください。

●EX-WINDOW用　LS_PATERN.X

マウスカーソルが残るというバグがありました。ソースリストの139行（PutArea2()の先頭）に，

```
MS_CURST( 0,0 );
```

を挿入して再コンパイルしてください。

●EX-WINDOW用　PP_COMPOSE.X

コマンド実行後，システムに復帰する方法が記述されていませんでした。復帰するには左右のマウスボタンを同時に押してください。

●EX-WINDOW用　EX_WIN.SYS

CHECKER PENの登録が間違っていました。実行ファイル名を，

```
CHECKER_PN.X
```

のように変更してください。

●SLASH用モデラMOD.X

コンパイル時に使用していたインクルードファイルの名前が違っていました。MOD.H内でインクルードしている，

```
__slashlib.h
```

を，

```
slashlib.h
```

に修正してください。ちなみに，

```
__tpl.h
```

はver.1.0のものを使用します。

リスト1

```
 49:   ▼8,Bitmap1 (552,8,590,40),0,0,0,0,sb7.pt4
 49:   ▼8,Bitmap1 (552,8,590,40),0,0,0,3006,

103:   ▼8,Bitmap2 (552,40,590,72),0,0,0,0,sb2.pt4
103:   ▼8,Bitmap2 (552,40,590,72),0,0,0,3001,

151:   ▼8,Bitmap3 (552,72,590,104),0,0,0,0,sb6.pt4
151:   ▼8,Bitmap3 (552,72,590,104),0,0,0,3005,

163:   ▼8,Bitmap4 (552,104,590,136),0,0,0,0,sb5.pt4
163:   ▼8,Bitmap4 (552,104,590,136),0,0,0,3004,

175:   ▼8,Bitmap5 (552,136,590,168),0,0,0,0,sb3.pt4
175:   ▼8,Bitmap5 (552,136,590,168),0,0,0,3002,

224:   ▼8,Bitmap6 (552,168,590,200),0,0,0,0,sb4.pt4
224:   ▼8,Bitmap6 (552,168,590,200),0,0,0,3003,

256:   ▼8,Bitmap7 (552,206,590,238),0,0,0,0,sbt1.pt4
256:   ▼8,Bitmap7 (552,206,590,238),0,0,0,3007,

267:   ▼8,Bitmap8 (552,238,590,270),0,0,0,0,sbt2.pt4
267:   ▼8,Bitmap8 (552,238,590,270),0,0,0,3008,

354:   ▼8,trash (524,234,548,262),0,0,0,0,tr.pt4
354:   ▼8,trash (524,234,548,262),0,0,0,2009,

373:   ▼8,Bitmap11 (20,72,534,169),0,0,0,0,1.pt4
373:   ▼8,Bitmap11 (20,72,534,169),0,0,0,256,

659:   ▼8,Bitmap53 (18,236,32,257),1,0,0,0,n32.pt4
659:   ▼8,Bitmap53 (18,236,32,257),1,0,0,1000,

666:   ▼8,Bitmap53 (48,236,62,257),1,1,0,0,n32p.pt4
667:   ▼8,Bitmap53 (80,236,94,257),1,2,0,0,n16.pt4
668:   ▼8,Bitmap53 (112,236,126,257),1,1,0,0,n16p.pt4
669:   ▼8,Bitmap53 (144,236,158,257),1,1,0,0,n8.pt4
670:   ▼8,Bitmap53 (176,236,190,257),1,5,0,0,n8p.pt4
671:   ▼8,Bitmap53 (208,236,216,257),1,6,0,0,n4.pt4
672:   ▼8,Bitmap53 (240,236,251,257),1,7,0,0,n4p.pt4
673:   ▼8,Bitmap53 (272,236,280,257),1,8,0,0,n2.pt4
674:   ▼8,Bitmap53 (304,236,315,257),1,9,0,0,n2p.pt4
675:   ▼8,Bitmap53 (336,236,344,257),1,10,0,0,n1.pt4
676:   ▼8,Bitmap53 (368,236,379,257),1,11,0,0,n1p.pt4
677:   ▼8,Bitmap53 (400,236,411,257),1,12,0,0,r32.pt4
678:   ▼8,Bitmap53 (432,236,443,257),1,13,0,0,r16.pt4
679:   ▼8,Bitmap53 (464,236,475,257),1,14,0,0,r8.pt4
680:   ▼8,Bitmap53 (496,236,507,257),1,15,0,0,r4.pt4

666:   ▼8,Bitmap53 (48,236,62,257),1,1,0,1001,
667:   ▼8,Bitmap53 (80,236,94,257),1,2,0,1002,
668:   ▼8,Bitmap53 (112,236,126,257),1,3,0,1003,
669:   ▼8,Bitmap53 (144,236,158,257),1,4,0,1004,
670:   ▼8,Bitmap53 (176,236,190,257),1,5,0,1005,
671:   ▼8,Bitmap53 (208,236,216,257),1,6,0,1006,
672:   ▼8,Bitmap53 (240,236,251,257),1,7,0,1007,
673:   ▼8,Bitmap53 (272,236,280,257),1,8,0,1008,
674:   ▼8,Bitmap53 (304,236,315,257),1,9,0,1009,
675:   ▼8,Bitmap53 (336,236,344,257),1,10,0,1010,
676:   ▼8,Bitmap53 (368,236,379,257),1,11,0,1011,
677:   ▼8,Bitmap53 (400,236,411,257),1,12,0,1012,
678:   ▼8,Bitmap53 (432,236,443,257),1,13,0,1013,
679:   ▼8,Bitmap53 (464,236,475,257),1,14,0,1014,
680:   ▼8,Bitmap53 (496,236,507,257),1,15,0,1015,
```

「PUSH BON!」のために作られた

外部関数の解説

Asakura Yuji 朝倉 祐二

X-BASICのみで作られた「PUSH BON!」ですが，多少，外部関数が拡張されています。大量のPCGデータを管理するためのものですが，リアルタイムゲームを作るために必要な関数も含まれていますので参考にしてください。

5月号の付録ディスク「こいのぼりPRO-68K」に収録された「PUSH BON!」。みなさん楽しんでいただけたでしょうか？早々に全ステージクリアした方もいることと思います。今月は予告どおり「PUSH BON!」で使用した外部関数の解説と，外部関数の作成方法を簡単に説明します。

PCGパターンの管理

遊んでもらえばわかるように，「PUSH BON!」はマイキャラ，ブロック，文字パターンなど，かなり多くのPCGパターンを使用しています。X68000はBGを使わないときで最大256個のPCGパターンを定義できますが，「PUSH BON!」で使用するPCGパターンは約700個にもなります。しかも「PUSH BON!」ではBGを2面とも表示しますので，定義可能なPCGの最大数は，さらに半減して128個になってしまいます。

しかし，パターンがいくつあろうとも1画面中に表示するパターンの数はそれほど多くはありません。タイトル画面だけで必要なパターンやゲーム画面だけで必要なパターンというのが必ずあるからです。PCGエリアを文字パターンなど，どの場面でも共通に必要なパターンを定義する領域と，場面に応じて表示に必要なPCGパターンを定義する領域を使い分ければ，限られたPCGエリアで大量のPCGパターンを扱うことができます。

write_chr()

いよいよ外部関数の解説に入りますが，X-BASICでPCGにパターンを定義をする，

SP_DEF()

と，スプライトを表示する，

SP_SET()

関数をよく知らない方は，まずX-BASICマニュアルを参照しながらこれからの説明を読んでいくと理解しやすいと思います。

先ほどいったとおり「PUSH BON!」はマイキャラのアニメーションパターンをたくさんもっていますが，これは128個のPCGエリアすべてを使用したとしても定義しきれない量です。まずは，キャラクタアニメーションのための外部関数を説明していきましょう。

たとえば10個のパターンでキャラクタアニメーションをさせるとき，あらかじめPCGに10個のパターンを定義しておき，表示のたびにSP_SET()で定義するパターンデータを変更する方法があります。セットするパターンデータを変更するだけなので，処理時間は速く，プログラムも簡単になりますが，アニメーションパターンの数が定義できるPCGエリアの大きさによって制限されてしまう欠点があります。そのため「PUSH BON!」のようにたくさんアニメーションパターンを使うゲームなどには向いていません。

これとは別に，PCGエリアのパターンデータを直接書き換えることによって，SP_SET()に指定するパターンデータを変えることなくキャラクタアニメーションをさせる方法があります。この方法はスプライトを表示したままの状態でPCGパターンを再定義するため，画面を乱さないためには垂直帰線期間中にパターンの書き換えを終わらせなければなりません。処理量は前者に比較するとかなり多くなりますが，キャラクタパターンの数がPCGエリアの大きさに制限されないため，実質上メモリのある限りアニメーションパターンを定義することができます。

「PUSH BON!」では大量のアニメーションパターンを使用するため，PCGエリアを直接書き換える後者の方法を取り入れています。PCGパターンの定義はX-BASICのSP_DEF()関数でもできますが，これでは処理が遅いかもしれないと思いましたので，マイキャラを定義しているPCGパターンエリアを変更する外部関数を作成しました。

write_chr()は「PUSH BON!」のマイキャラパターン変更・表示を行う外部関数です。「PUSH BON!」ではマイキャラをスプライト番号4～7にPCGパターン番号78,79,94,95を定義して表示しています。SP.FNCは外部関数のプログラム部分はもちろん，そのほかに約700個のPCGパターンデータを含んでいます。write_chrはpatとdirの値からPCGに定義するSP.FNC内のパターンデータの位置を割り出し，パターン番号78,79,94,95のパターンを書き換えたあとに，スプライト番号4～7のスプライトスクロールレジスタを書き換えてマイキャラの表示位置を変更します。引数の詳細は表1を参考にしてください。

write_pcg()

write_chr()はマイキャラパターンを定義しているPCGだけにパターンを定義する外部関数でしたが，write_pcg()は128個のPCGパターン番号の好きな場所にSP.FNC内のパターンを定義することができます。

タイトル画面で使用するPCGパターンには「PUSH BON!」のタイトル文字を定義しています。これはBGにタイトル文字を表示するためのものです。実は当初はタイトル文字だけをラスタスクロールさせて遊ぶつもりだったので，わざわざBGにタイトル文字を表示するように変更したのですが，結局はほかの部分の調整に手一杯でラスタスクロールに手をつけることができず，残念に思っています。

ontime()

ゲーム中の画面右下にはマイキャラのアニメーションが常に表示されています。こ

のアニメーションパターン，夜に遊んでいる人はマイキャラが本を読んでいるものしか知らないと思いますが，遊んでいる時間によって数種類のアニメーションパターンがあります。たまには普段と違う時間帯で遊んでみるのも楽しいと思いますよ。

さて，このアニメーション，開発中は約1秒ごとにパターンを書き換えていたのですが，それでは間隔が開きすぎるという意見があったので，ディスク収録版では約0.7秒ごとにパターンを書き換えるようにプログラムしています。1秒単位ならX-BASICならTIME$を使って時間の管理ができたのですが，1秒未満となるとお手上げです。

ontime()はX-BASICで1/100秒単位で時間管理を可能にする外部関数です。

```
int a,b
a=ontime()
  ∫
 処理A
  ∫
b=ontime()
```

のように使うと，aとbにX68000が起動してからの時間が1/100秒単位で格納されます。また厳密には違いますが，b−aの値が処理aに費やした，おおよその処理時間を表します。

vdisp()

X-BASICには垂直帰線期間を検出する命令がありませんので，垂直帰線期間を検出する命令を外部関数で作成しました。

```
vdisp()
```

これは垂直帰線期間を検出すると，処理を終える外部関数です。スプライトやグラフィック画面をスクロールさせるプログラムをコンパイルして実行すると，垂直表示期間中にスプライトスクロールレジスタやグラフィックスクロールレジスタの書き換えが起こり，画面が乱れることがあります。このような症状が出た場合は，スクロールさせる直前にvdispで垂直帰線期間を検出しておけば，表示の乱れはなくなります。

注）「PUSH BON!」のために作成した以上の外部関数は，私の手抜きにより引数のエラーチェックをほとんどしていません。使用の際は十分に注意してください。

外部関数の作成

最後におまけディスクに収録されたSP.Sを利用して外部関数を作成する方法を説明します。アセンブラの知識が最低限必要ですから，外部関数を作ろうという人は，まずアセンブラの勉強から始めてください。

外部関数は大きく分けて，外部関数のコマンド名や引数を定義する5つのテーブル部分と，外部関数の処理を記述するプログラム部分で構成されます。

★インフォメーションテーブル

SP.Sでは18〜41行がインフォメーションテーブルです。以下に紹介する各テーブルの先頭アドレスなどを定義します。ここは特別なことを必要としない限り変更する必要はありません。詳しい説明が知りたいなら，XCプログラマーズマニュアルを参照してください。

★トークンテーブル

SP.Sの43〜48行がこの部分です。ここには外部関数のコマンド名を定義します。SP.Sのようにひとつの外部関数ファイルに，複数の外部関数を含む場合，このように外部関数名を0で区切って複数定義します。トークンテーブルの終わりには“0”を2つ続けて書きます。

★パラメータテーブル

トークンテーブルで定義した外部関数が引数を必要とするか，引数の数はいくつかなどの情報を定義している先頭アドレスを指定します。SP.Sでは52〜56行の部分です。ここはトークンテーブルと1対1で対応しており，たとえばトークンテーブルで先頭のontime()の引数情報の先頭アドレスはontime_par，同様にwrite_pcg()はwrite_pcg_parに……というようになります。

★パラメータIDテーブル

引数の型や戻り値など外部関数のパラメータ全体の内容を定義する部分です。57〜77行ですが，定義可能なパラメータIDの詳細はXCプログラマーズマニュアルを参考にしてください。

★実行アドレステーブル

78〜82行で各外部関数のエントリアドレスを指定します。ここはトークンテーブルで定義した外部関数名に対応させて，エントリアドレスを順番に並べます。

以上で外部関数の作成の50%を説明しました。残りの50%は引数と戻り値の受け渡し方法の説明になりますが，XCプログラマーズマニュアルでも6ページにわたって説明されているものです。残り少ないスペースですので，最低限必要なことだけ説明します。

外部関数エントリアドレスに処理がきた時点で，

　　SP+8　　1番目の引数の先頭アドレス

表1　外部関数リファレンス

★vdisp()
機能
　垂直帰線期間を待つ
引数
　なし
戻り値
　void
★ontime()
機能
　システムの起動時間を1/100秒単位で返す
引数
　なし
戻り値
　int
★write_chr(x,y,pat,wait,dir,h)
機能
　PCGパターン番号78,79,94,95のパターンをpatで指定のパターン番号で書き換える
引数
　x：スプライトX座標
　y：スプライトY座標
　pat：PCGパターン番号（表2参照）
　wait：垂直帰線期間を待つ回数（1/60秒単位）
　dir：キャラクタの向き（表2参照）
　h：水平反転表示をしない…0
　　　〃　　　　する……1
戻り値
　void
★write_pcg(pcg_pat,pat,wait)
機能
　pcg_patで指定のpcgパターン番号のパターンをpatで指定のパターン番号で書き換える
引数
　pcg_pat：0〜127
　pat：0〜148　タイトル画面で使用するパターン
　　149〜340　ブロックパターンなどゲーム中に使うパターン
　　341〜612　マイキャラアニメーションパターン
　　613〜703　エンディングで使用するパターン
　wait：垂直帰線期間を待つ回数（1/60秒単位）
戻り値
　void

表2　マイキャラクタのパターン番号

dir=0:キャラクタは正面向き
pat=0〜7
dir=1:キャラクタは背面向き
pat=0〜7
dir=2:キャラクタは右（左）向き
pat=0〜7（左向きのときはh=1）
dir=3:ギブアップ時などのパターン
pat=0〜3
dir=4:画面右下のキャラクタパターン
pat=0〜7

SP＋18　2番目の引数の先頭アドレス
SP＋28　3番目の引数の先頭アドレス
　：　　　　：

以下同様に，10バイトおきに引数の先頭アドレスが繰り返し現れます。また引数の型と有効データ長は，

float型　8バイト
int型　下位4バイト
char型　下位1バイト
str型　下位4バイトがポインタ

となっています。

外部関数の戻り値はレジスタを使って渡します。正常終了の場合d0レジスタに0を入れ，戻り値を必要とする関数なら戻り値を格納したエリアの先頭アドレスをa0レジスタにセットして終了します。戻り値のエリアは10バイト必要で，先頭の2バイトは必ず0，残りの8バイトに戻り値を格納します。引数の型と有効データ長の関係は上と同様です。

異常終了の場合は，d0レジスタに0以外の値をセットして，a1レジスタにエラーメッセージを格納した先頭アドレスを指定します。

反省と今後について

駆け足で外部関数の説明と作成方法を説明しました。外部関数の作成方法については本当に大ざっぱにしか説明していないので，わからない部分が多いことと思います。説明が足らない部分はソースを読めば少しはわかってもらえると思いますが，ソースも汚くてあまり参考にならないかもしれませんね（反省）。

外部関数の作成方法について詳しく知りたい人が多いようならば，別の機会にもっと誌面をたくさんいただいて，外部関数を含んだプログラムをコンパイルした際に，必要になるライブラリの作成方法を交えて説明したいと思います（ライブラリはX-BASICと引数の受け渡し方が違うだけですから，作成は難しくありません）。

最後に私自身の今後の予定ですが，力不足ながらゲームはもちろん，システム関係も手がけていきたいと思っていますので，今後とも応援をよろしくお願いします。それではまた。

PUSH BON!オリジナル面データ大募集

5月号の特別付録「こいのぼりPRO-68K」に収録された「PUSH BON!」。僕もお絵書き人として参加したゲームでしたので，読者からの反響が非常に気になっていました。

現在，続々と返送されてくるアンケートハガキをチェックしているのですが，読者からの反響は上々。多くの皆様に楽しく遊んでもらっているようで，非常にうれしいですね。すでに全ステージをクリアして，オリジナルステージを作って遊んでいる人もいるようです。

そこで，せっかくステージコンストラクションがあることだし，読者からのオリジナルステージを募集します。お送りしていただくものは，コンストラクションで作成したステージデータが収録されているファイル，

PUSH_USER!.MAP

そして，ベストステップが格納されているレコードファイル，

PUSH_BON!.REC

の2ファイル。条件としては，絶対にクリアできるものにかぎります。

なお，応募ステージ数に制限はありません。入魂の一作，1日1ステージ対応特大365ステージでもOKです。そして，応募先は「PUSH BON!」オリジナルステージ係，応募締め切りは7月31日とします。で，送られてきたステージデータの中で優秀なものは，次回の付録ディスクに収録させていただきます。さらに，応募数が多ければ，部門賞を設け，原稿料のほかに副賞もあげちゃいましょう。

では，奇抜なステージ，100ステップもかかるようなウルトラC級のステージ，こだわりのある美しいステージ，とにかくオリジナリティあふれるステージデータをお待ちしています。

でも，オリジナルステージを作るのはいいけど，どうやったらいいのかよくわからない人もいるでしょうから，ここでステージデータの作成のポイントをお教えしましょう。それは，スターブロックを適当に配置して，スターブロックの道筋を考えながらそのほかのブロックを配置していくのです。ひとつ固定ブロックを置くだけでずいぶんと解法が変わりますから，とにかくいろいろ試してみてください。

さて，ひとしきり盛り上がったところで，ゲームに関してちょっとお話しておきましょう。

まず，クリアするのにだいぶ苦戦している人がいますが，絶対に解けないステージは存在しません。一応，ステージデータを作りながらクリアチェックしたので，おかしなデータは紛れ込んでいないはずです。必ずクリアできると信じてがんばってください。ただし，解法を教えてくれ，といわれても現在クリアできる自信はありません（なにしろ2日で50面分のデータを一気に仕上げたので覚えていない）。

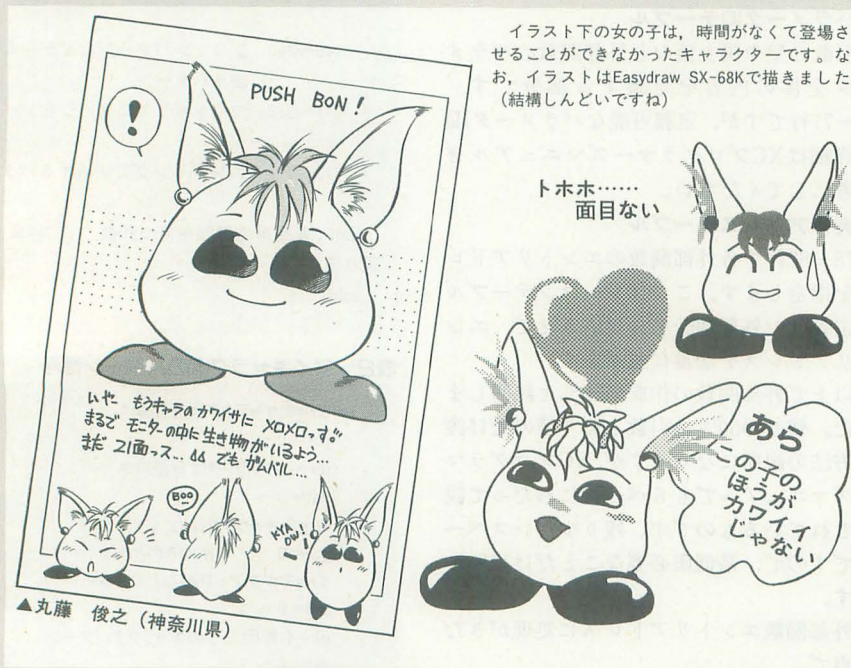
で，僕はお絵書き，ステージ作成のほかに，デザイン関係も担当させてもらいました。そのときにひとつだけ注意した点があります。

それは，画面に変化をもたせることです。ゲームの性格上，ステージをクリアしても結局同じような画面が続きます。そこで，ゲーム画面の右下にキャラクターを配置して，時間ごとに違うアニメーションをさせたり，ゲームプレイデモを用意したり，音楽を10面ごとに変化させたりしています。マイキャラのアニメーションもそうですね。とにかくプレイヤーを飽きさせない，ということを念頭に置きました。

最後に，丸藤さんとってもかわいいイラストありがとう。はっきりいって僕が描いたものよりも数段かわいいです。「STUDIO X」宛に送られてきたものを，思わず横取りしてしまうほどかわいいんですもの。やっぱり，画力のある人は違うなあ（ちょっと……いや，かなり悔しい）。僕も精進せねば，ということで，また，朝倉氏と一緒にゲームを作るつもりです。うん，がんばらなくちゃ。

（浜崎正哉）

イラスト下の女の子は，時間がなくて登場させることができなかったキャラクターです。なお，イラストはEasydraw SX-68Kで描きました（結構しんどいですね）

▲丸藤 俊之（神奈川県）

Oh!X LIVE in '94

©CAPCOM「スーパーストリートファイターII」より

X68000・Z-MUSIC+
PCM8用

キャミィのテーマ

Hirata Yasutoshi
平田 恭敏

X68000・Z-MUSIC+
ZPP.X用(SC-55mkII対応)

The End of Love

Torigoe Eiji
鳥越 英司

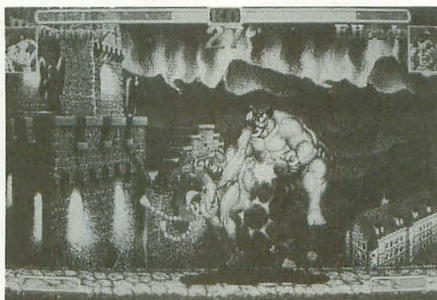
移植希望を込めて全キャラ制覇，なんていっていた「ストII」ですが，X68000やゲーセンでみんなが熱中しているあいだに，新キャラ・キャミィのテーマが送られてきました。やっぱり女の子が優先かな？ もう1曲はオリジナルのすてきな作品です。

新たなる戦士登場！

さて，今年のGAME OF THE YEARで賞を取り損ねた作品として(？)記憶に新しい「ストリートファイターII」ですが，とにもかくにもX68000シリーズにも無事に完全移植されたのは喜ばしい限りですよね。ストIIの作品が掲載されるたびに「移植して」と熱烈なラブコールを送ったかいがあったというものです。し・か・し，すでにさらに進化した「スーパーストII」が登場，ゲーセンでは定番と化しているのでした。そのうえいまでは「スーパーストIIX」が猛威をふるっていますね。「スーパー」と名乗ってから，新キャラや新ワザも増えました。ということで，新キャラ「キャミィのテーマ」をお届けしましょう。X68000にも移植されるといいですねぇ。

演奏にはPCM8.XとZ-MUSICシステムが必要です。

内蔵音源のみで再現しているにもかかわらず，かなりいい仕上がりですね。思わずかけ声(？)を入れたくなってしまいます。曲は無限ループになっています。

作者の平田君は，この曲について2つの作品を送ってくれました。掲載されたバージョンでは，AD PCMを多用して，ギター

スーパーストリートファイターII

の音質を上げているようです。このような試みも面白いですよね。リストも短めですし，こりゃー入力するっきゃないですね。

あ，それから平田君，大学ご入学おめでとうございます。

愛の終わりに……

さて，2曲目は久しぶりにオリジナル曲はいかがでしょう。LIVE inではどんな曲でも募集しています。作曲家を目指すあなたにも掲載のチャンスはあるのです。

タイトルは「The End of Love」。洒落たタイトルつけてくれるじゃないの，まったく。作品のほうも負けず劣らずお洒落にキマっています。ちょっとヘヴィなベースのフレーズから始まるフュージョン系のサウンドがばっちり決まっています。

演奏にはZ-MUSICシステム，SC-55同等品，そしてZPP.Xが必要です。正確にはSC-55mkIIに対応していますので，それ以外の音源をお使いの方は116ページのコラムを読んでリストを修正してくださいね。ZPP.Xは入力の効率を上げるために開発されたもので，1993年10月号の付録ディスク「秋祭りPRO-68K」に収録されています。これを使うとリストを短くすることができます。実際にエキスパンドされたリストは，掲載されたものより2割ほど大きくなっています。この作品では繰り返しなどはあまり多用していませんが，ちゃんと効果を発揮しているのがわかりますね。

かなりかっこよく仕上がった作品ですが，ピアノのフレーズはひっかかりをつけすぎのようです。もう少しスムーズでもよかったかもしれません。意識的にしたのでしょうが，手弾き感覚からはずれてしまいます。

曲の感じからすると大人の恋が終わったというところでしょうか。最後の風の効果音がいいですね。

この作品も聴かないとソンすると思うな。ほかにもオリジナル曲を送ってくれる常連さんたちがいますが，みんなは，入力して感想を送るんだぞ。感想もらうって嬉しいでしょ。オリジナル曲だと思い入れも強いし。では，また来月。 (SIVA)

リスト1 キャミィのテーマ

```
 1: .COMMENT "SUPER STREET FIGHTER II"  ~ Cammy ~ (C) CAPCOM
'94/ 3/ 3     By WATASHIDA!
 2: .ADPCM_BLOCK_DATA = CAMMY2.ZPD
 3: /  For  Z-MUSIC   Ver. 2.00         < OPM + PCM8.X >
 4:
 5: (i)
 6: (b0)
 7: (o140)
 8:
 9: (m1,2000)(aFm1,1)
10: (m2,2000)(aFm2,2)
11: (m3,2000)(aFm3,3)
12: (m4,2000)(aFm4,4)
13: (m5,2000)(aFm5,5)
14: (m6,2000)(aFm6,6)
15: (m7,2000)(aFm7,7)
16: (m8,2000)(aFm8,8)
17: (m9,2000)(aAdpcm,9)
18: (m10,2000)(aAdpcm,10)
19: (m11,2000)(aAdpcm,11)
20: (m12,2000)(aAdpcm,12)
21: (m13,2000)(aAdpcm,13)
22: (m14,2000)(aAdpcm,14)
23:
24:
25: /   AR D1R D2R  RR D1L  TL  RS MUL DT1 DT2 AME   G.Piano
26: (@1,          31,  5,  2,  1,  1, 35,  1,  1,  7,  0,  0
27:       25,  5,  2,  1,  1, 45,  2, 13,  0,  0,  0
28:       31,  5,  2,  1,  1, 38,  1,  3,  3,  0,  0
29:       20,  7,  2,  6, 10,  0,  2,  1,  0,  0,  0
30: /   CON FBL SM
31:       2,  6, 15)
32:
33: /   AR D1R D2R  RR D1L  TL  RS MUL DT1 DT2 AME    Guitar
34: (@2,          16, 15,  1,  3,  3, 26,  0,  3,  3,  0,  0
35:       19, 31,  1,  3,  0, 27,  0,  1,  0,  0,  0
36:       26, 31,  1,  3,  0, 26,  0,  1,  5,  0,  0
37:       27, 31,  1,  8,  0,  0,  0,  1,  0,  0,  0
```

```
 38: /  CON FBL  SM
 39:      0,  7, 15)
 40:
 41: /   AR D1R D2R  RR D1L  TL  RS MUL DT1 DT2 AME      Sax
 42: (@3,          29,  4,  0,  6,  1, 27,  1,  1,  6,  0,  0
 43:      29, 15,  0,  8,  5, 39,  1,  8,  0,  0,  0
 44:      29, 31,  0,  3,  0, 37,  1,  1,  5,  0,  0
 45:      16, 31,  0, 10,  0,  1,  1,  2,  0,  0,  0
 46: /  CON FBL  SM
 47:      2,  7, 15)
 48:
 49: /   AR D1R D2R  RR D1L  TL  RS MUL DT1 DT2 AME      BASS
 50: (@4,          31,  5,  5,  6,  5, 25,  0,  1,  7,  0,  0
 51:      18,  6,  3,  6,  5, 50,  0,  0,  0,  0,  0
 52:      31,  5,  3,  6,  3, 30,  0,  0,  3,  0,  0
 53:      31,  3,  3,  8,  3,  0,  0,  1,  0,  0,  0
 54: /  CON FBL  SM
 55:      1,  5, 15)
 56:
 57:
 58: /  M E L O D Y
 59:
 60: (T1) [K.SIGN -C,-D,-E,-G,-A,-B,]  @Q4 @3 @M4 L8
 61: (T1) [DO] V15 |:8R1:| O4
 62: (T1) F2R>B<DE F4 GF4.D4 E2 Q6E.F.G Q8 D.(AB16)B8AG.Q5F.G @Q6
 63: (T1) F4.Q8 G16F16 @Q6E4.>B Q6<D.F.BB.<C.>B @Q6A4.G4.F4
 64: (T1) F4G16..&(GA64)&A4GF4  ~~F2&|:15_F32&:|F32  |:5R1:|
 65: (T1) V15 B4BA4A4D D4EF4DE4 GFEDDEFE& EE&E4.EDE
 66: (T1) EFG<D4>B<DD& DEE2.&E2..&|:4__E32&:| R32R2.... R1 R1 [LOOP]
 67:
 68:
 69: (T2) [K.SIGN -C,-D,-E,-G,-A,-B,]  @Q3 @3 L8
 70: (T2) R16 [DO] |:8R1:| O4 V14 P1 @K6
 71: (T2) F2R>B<DE F4GF4.D4 E2 E.F.G  D.B.AG.F.G
 72: (T2) F4.G16F16E4.>B <D.F.BB.<C.>B  A4.G4.F4 F4GA4GF4
 73: (T2) @2 O4 P1 @B120
 74: (T2) F2R>B<DE F4GF4.D4 E2E.F.G D.B.AG.F.G
 75: (T2) F4.G16F16E4.>B <D.F.BB.<C.D>
 76: (T2) @3 @V120 @K6 B4BA4A4D D4EF4DE4 GFEDDEFE& EE&E4.EDE
 77: (T2) EFG<D4>B<DD& DEE2.&E2..&|:4__E32&:| R32R2.... R1 R1 [LOOP]
 78:
 79:
 80: (T3) [K.SIGN -C,-D,-E,-G,-A,-B,]  R8 @Q1 [DO] |:8R1:|
 81: (T3) L8 O4 @K-4 @3 V13 P2 F2R>B<DE F4GF4.D4 E2E.F.G D.B.AG.F.G
 82: (T3) F4.G16F16E4.>B <D.F.BB.<C.>B A4.G4.F4 F4GA4GF4
 83: (T3) @2 @K-8 P2 O4
 84: (T3) F2R>B<DE F4GF4.D4 E2E.F.G D.B.AG.F.G
 85: (T3) F4.G16F16E4.>B <D.F.BB.<C.D16 P3 @V120 D4DC4C4>B F4GA4FGA
 86: (T3) BAGF4GAF& FE&E4.EFG GABA4GFD& DEE2.&E2...&|:4__E32&:|
 87: (T3) R32R2.... R1 R1 [LOOP]
 88:
 89:
 90: (T4) [K.SIGN -C,-D,-E,-G,-A,-B,]  [DO]
 91: (T4) @2 V15 @K3 O4 L8 |:16 R1:|
 92: (T4) F2R>B<DE F4GF4.D4 E2 Q7E.F.G Q8D.(AB16)&B8AG.F.G
 93: (T4) F4.G16F16E4.>B <D.F.BB.<C.D ~D4DC4C4>B F4GA4FGA
 94: (T4) BAGF4GAF& FE&E4.EFG GABA4GFD&
 95: (T4) DEE2.&E2..&|:4__E32&:|  R32R2.... R1 R1 [LOOP]
 96:
 97: /  C H O R D
 98:
 99: (T5) [K.SIGN -C,-D,-E,-G,-A,-B,]  @1 L8 [DO]
100: (T5) @V120 @Q3 |: B.B.B4BA4 B.G.E4BA4 :|
101: (T5) V14 RG16F16R16G.G.B.E RDF.F16F.G.A
102: (T5) RB16G16B.>B16<F.G.B A.<D.D> ~A.<~D.> Q8~~A @Q6
103: (T5) V14 R |:4 G.G.G.G.G4  F.F.F.F.F4  A.A.A.A.A4  G.G.G.G.G4:|
104: (T5) G.G16A2B R4B4B.BB. RB.B16<D2D& DEE2.
105: (T5) R>B2<D2 D4.E4.>A4.A4<D4>A4 B4.A&A2 [LOOP]
106:
107:
108: (T6) [K.SIGN -C,-D,-E,-G,-A,-B,]  @1 L8 [DO]
109: (T6) @V120 @Q3 |: G.G.G4GF4 G.E.C4GF4 :|
110: (T6) V14 RE16D16R16E.E.G.C >RB<D.D16D.E.F
111: (T6) RG16E16G.>G16<D.E.E F.A.F ~F.~~A. Q8~~F @Q6
112: (T6) V14 R |:8 E.E.E.E.E4  D.D.D.D.D4 :|
113: (T6) E.E16F2G R4G4G.GG. RG.G16A2A& ABB2.
114: (T6) RG2A2 B4.<C!4.>F4.F4A4F4 G4.F&F2 [LOOP]
115:
116:
117: (T7) [K.SIGN -C,-D,-E,-G,-A,-B,] @1 L8  [DO]
118: (T7) @V120 @Q3 |: E.E.E4ED4 E.C.>G4<ED4 :|
119: (T7) V11 RR32E16D16R16E.E.G.C >RB<D.D16D.E.F
120: (T7) RG16E16G.>G16<D.E.E F.A.FF.A.F16. @Q6
121: (T7) V14 O3 R |:4 |:B.B.B.B.B4:| < C.C.C.C.C4 > B.B.B.B.B4 :|
122: (T7) O4 C.C16D2E R4E4E.EE. RE.E16F2F& FGG2.
123: (T7) RE2F2 G4.A4.D4.D4F4D4 E4.D&D2 [LOOP]
124:
125: /  B A S S
126:
127: (T8) [K.SIGN -C,-D,-E,-G,-A,-B,] @4 V13 L8 Q7 [DO]
128: (T8) O3 |:E.B.<E4D>A4 <C.>G.E<C.C.D :|
129: (T8) O2 |: A.A.A :| |: B.B.B :| < |: C.C.C :| DD>A<AF.G.A
130: (T8) O3 |:4 E<E16D16>EEE.G.E  >BBR16B16BB.<F.D
131: (T8)          >AAAAA.<C16>A<E  DDDDD.E.F  :|
132: (T8) O2 A<A>AB4<B>B<C& |:8C:|  C<C>CD4<D>DE& |:8E:|
133: (T8)     E<E>EF4<F>FG& G<G>G>AAAA<D R|:7D:|  CCCDRDDD [LOOP]
134:
135:
136: /  D R U M S
137:
138: /  BASS
139: (T9) O0 L4    [DO]
140: (T9) CCCC CCC8C16C16&C4 CCCC CCC2
141: (T9) |:3CCCC:| C16C8C16C16C8C16C8C16C8.C8
142: (T9) |:16CCCC:|
143: (T9) C4.C2C8 &C8C4C8C8.C&C16 C4.C2C8 &C8C4C8C8.C&C16
144: (T9) C4.C2C8 &C8C4C8C8.C8.C8 &C8C.C8C.C.C8&C2     [LOOP]
145:
146: /  SNARE
147: (T10) O0 L2   [DO]
148: (T10) R1 RR16D8.D8D16D16
149: (T10) R4DDD4R16D16D16D16D16D8. R4D2D2D2D2D2D2D4D8.D8.D8
150: (T10) R4|:6D:|D4&D16D8.D8D16D16 &D4|:6D:|D4&D16D8.D4
151: (T10) R4|:6D:|D4&D16D8.D8D16D16 &D4|:6D:|D4&D16D16D8.D16D16D16
152: (T10) |:R8D16D16&D4.D8D16D8. &D4D2D8D16D16:|
153: (T10) R8D16D16&D4.D8D16D8. &D4D2D4 &D4D2D4. D8D4.D8E8F8 [LOOP]
154:
155: /  TAMTAM   E -----> HIGH  :  F ----->LOW
156: (T11) O0 L16 R1  [DO]
157: (T11) R1R1R2..EF R1R1R1R4.EF&F2
158: (T11) |:7R1:| R2..E16F16&F1 |:7R1:|
159: (T11) V10|:3R4F2.R1:| V9 R1R2R8R16D8E8F16&F1  [LOOP]
160:
161: /  SYMBAL
162: (T12) O0 L1 [DO]
163: (T12) |:3GR:| GR2G8.G8.G8
164: (T12) V10|:4GRRR:|
165: (T12) |:G4.G2G8 &G1:| G4.G2G8 &G2.. V9 G1 G8G4.G2&G8 [LOOP]
166:
167: /  H.H.  A ---> OPEN : B -----> CLOSE
168: (T13) O0 L16 [DO]
169: (T13) |:R4|:7BBA8:|:|  L8 |:14BA:|B8.B8.B8
170: (T13) |:64BA:|
171: (T13) |:3R1 R4ABABAB:| RBABABAB R2RABA  [LOOP]
172:
173: /            GUITAR II
174:
175: (T14) [K.SIGN -C,-D,-E,-G,-A,-B,]  [DO] O2 L2.
176: (T14) |: ED4 CD4 :|  L1
177: (T14) > A B <C  D4.>B8<D8.E8.D8  E2R2
178: (T14) |:7R:|
179: (T14) |: E >B <A D :|
180: (T14) L8  C4.D2E &EEE16 EE16 EE E16 E16 E
181: (T14) E4.F2G &GGG16 GG16 GGG16 G16 G
182: (T14) E4.A2G &GGGA4FF D4DD16 DD16 DDD16 D16 DC4.D2&D   [LOOP]
183:
184: (P)
```

リスト2 キャミィのテーマの音色コンフィグファイル

```
.O0C = SLDK.pcm,V85
.O0D = RVBS2.pcm,V80
.O0E = DRT2.pcm,V75
.O0F = DRT1.pcm,V80
.O0G = CRSH1.pcm
.O0A = HO1.pcm,V75
.O0B = CH1.pcm,V70
.O1A-= DSG_O2C.pcm,P-4,V50
.O1B-= DSG_O2C.pcm,P-2,V50
.O2C-= DSG_O2C.pcm,P-1,V50
.O2D-= DSG_O2C.pcm,P+1,V50
.O2E-= DSG_O2C.pcm,P+3,V50
.O2F = DSG_O2C.pcm,P+5,V50
.O2G-= DSG_O2C.pcm,P+6,V50
.O2A-= DSG_O2C.pcm,P+8,V50
```

リスト3 キャミィのテーマのカウンタ表示

```
 1:00000000 00001800   2:0000000C 00001800   3:00000018 00001800   4:00000000 00001800
 5:00000000 00001800   6:00000000 00001800   7:00000000 00001800   8:00000000 00001800
 9:00000000 00001800  10:00000000 00001800  11:000000C0 00001800  12:00000000 00001800
13:00000000 00001800  14:00000000 00001800
```

リスト4 The End of Love

```
 1: .comment  T h e   E n d   o f   L o v e    by Eiji Torigoe
 2:
 3: (i)
 4: (b1)
 5:
 6: .roland_exclusive $10,$42={$40,$0,$7f,$0}
 7: .sc55_reverb $10={4,4,0,94,60,30,40}
 8: .sc55_chorus $10={3,0,46,14,40,20,20,10}
 9: .sc55_v_reserve $10={2,4,4,6,3,3,0,0,0,4,0,0,0,0,0,0}
10:
11: (m1,3000)(amidi1,1)
12: (m2,3000)(amidi2,2)
13: (m3,3000)(amidi3,3)
14: (m4,3000)(amidi4,4)
15: (m5,3000)(amidi5,5)
16: (m6,3000)(amidi6,6)
17: (m10,3000)(amidi10,10)
18: (m11,3000)(amidi10,11)
19: (m12,3000)(amidi10,12)
20:
21: (t1)  @i$41,$10,$42 @e110,20
22: (t2)  @i$41,$10,$42 @e100,16
23: (t3)  @i$41,$10,$42 @e100,16
24: (t4)  @i$41,$10,$42 @e110,8
25: (t5)  @i$41,$10,$42 @e120,24
26: (t6)  @i$41,$10,$42 @e120,8
27: (t10) @i$41,$10,$42 @e80,44
28: (t11) @i$41,$10,$42 @e80,44
```

▶あーあ，「LIVE in'94」に投稿するはずのファイルに間違って上書きして消してしもーた！ 打ち込んだ日々が，全部消えた。悲しんでもしかたねーからなにか作曲して投稿するぞ！
板橋　芳則(19)福島県

```
 29: (t12) @i$41,$10,$42 @e80,44
 30:
 31: (o108)
 32:
 33: ////////////////////
 34: #ext_on
 35: /---------- bass
 36:
 37: (t1) @37@v114@u127o1 @L8
 38: (t1) r2r8(g8<g)&(g4>g)
 39: (t1) [a1][a2][a1][a2][a3][a4]
 40: (t1) [a][a][a][a7]
 41: (t1) [a][a][a8][a9]
 42:
 43: *a1* |: a8rra<g&ga>g&ga rrag&ga<ga>g(a8,g),8 :|
 44: *a2* |: f8rrf<e&ef>e&ef rrfe&ef<fg>|e(f8,e) :| fg8
 45: *a3* d8rrd<c&cd>c&cd rrdc&cd<cd>c(d8,g)
 46: *a4* |:3 g&gg |r4 :| r8_2
 47: *a5* f8rrf<e&ef>e&ef rrfe&ef<fg> e(f8,e)
 48: *a6* g8rrg<f&fg>f&fg rrgf&fg<ga> f(g8,f)
 49: *a7* (a4,f+) r2g4 a1&a2 r8(a8<a)&(a4>a)
 50: *a8* r*1536
 51: *a9* ¥-16 i3@123@v120o3 a*1920
 52: *a* [a1][a2][a1][a5][a6]
 53: **
 54: /---------- st L
 55:
 56: (t2) @50@v50@u125o3 L4 p1@k2
 57: (t2) r1
 58: (t2) [b1][b2][b1][b2][b3][b4]
 59: (t2) [b][b][b][b6]
 60: (t2) [b][b][b7][b8][b9]
 61:
 62: *b1* |: 'a2<eb<c''a2<ea<c' :|
 63: *b2* |: 'f2<cga'|'f2<cfa' :| 'f<cfa''f<cga'
 64: *b3* 'd2a<ef''d2a<df'
 65: *b4* @49|:3 '<g<cdg'16'<g<cdg'8 |r4 :| r8 @50
 66: *b5* 'f2<cga''f2<cfa' 'g2<dgb''g2<dgb'
 67: *b6* r1'a<c+ea'384
 68: *b7* @49@L384 ¯10 @m30@s4
 69: *b8* |: 'a<e<c' 'f<ca' :|
 70: *b9* ¥10[b8]
 71: *b* [b1][b2][b1][b5]
 72: **
 73: /---------- st R
 74: /
 75: (t3) @50@v50@u125o3 L4 p2@k-2 r*8
 76: (t3) r1
 77: (t3) [b1][b2][b1][b2][b3][b4]
 78: (t3) [b][b][b][b6]
 79: (t3) [b][b][b7][b8][b9]
 80:
 81: /---------- p
 82:
 83: (#A,"@u76")
 84: (#B,"@u127")
 85: (#C,"@u114")
 86: (#D,"@d4")
 87: (#E,"@d0")
 88: (#X,"L8")
 89: (#Z,"@L18")
 90:
 91: (t4) i8@4@v127o4 X @p38
 92: (t4) r1
 93: (t4) r*1824 r A d+32 B e16.ga
 94: (t4) A b32 B 'e<c'42'd4b''>b4<g''>g<e''ca'
 95: (t4) C r*16'a<c''a<c'8 'gb''gb'16'fa''eg'8'fa'
 96: (t4) z127,,97,,117,127 D {age}{d+dc}
 97: (t4) B |:|: >f*16<'cd+'>f*8<|'cf':|'cf'16>f*8<
 98: (t4) 'cf'16'cd+'32:| E
 99: (t4) D >a< C {deg} B a*16<c>a*8<
100: (t4) z127,112,97@d3 {d+dc}>{ag+g}{d+dc}d*16c*8 E
101: (t4) Z z126,100,126 'b<e''b<g''b<d+'12'b<d''b<c+''b<c'12
102: (t4) 'ea''eg+''eg'12>'a<d+''a<d''a<c'12 X
103: (t4) @u122 D a'<cf'a*16'<cg'a*8 '<cf'a*16'<ce'a*8'<cf'
104: (t4) B 'b<g' C <{deg} z127,114,114 a*16gg*8
105: (t4) B 'b<g'18'a<f'18'g<e'12 'g<d'16>'b<g'32
106: (t4) r*1536 X<
107:
108: (t4) |:|: E 'ca''ca'|:3{rr'>b<g'}:|'ca''ca''>b<g' :|
109: (t4) D f{def}g*18f*18e*12f4 |>'b<g''b<e' ED
110: (t4) <'cf''cf'{rrg}{def}'ce'18'cf'18'cg'12'g<d'{bge} :|
111: (t4) 'ce''cf' >'b<g'18'b<d'18'b<e'12 'b<g'18'b<a'18'b<g'12
112: (t4) <'g<d''g<c''db''dg'
113:
114: (t4) ZE <'ca'>'b<g''g<e'12 'f+<d+''f<d''e<c'12
115: (t4) 'ca'>'b<g''g<e'12 'f+<d+''f<d''e<c'12
116: (t4) 'c+ea'384
117:
118: (t4) r1r2L2o5_36 D 'e4<c''d4b' 'db''c1a''c4a''d4b'
119: (t4) 'e1<c''f<d''db' 'db''c1a' '>b<g'
120: (t4) _4'c1a''db''e<c' 'g<e'180
121: (t4) z94,127 'f16<d''f<d'84 'e16<c''e1<c'
122: (t4) B 'd4b''f4<d''e<c''ca' 'ca''>b<g' 'f<d''db'
123:
124: (t4) i0@49@v80o4 X p3 E
125: (t4) [d1] ¥10[d1]
126:
127: *d1* |:|: '>a<c''ce''a<c''ce''gb''ce''b<d''ce' :|
128:        |: '>fa''>a<c''fa''>a<c''eg''>a<c''gb''>a<c':|:|
129: **
130: /---------- g
131:
132: (#F,"z127,127,107")
133: (#G,"z127,107,127")
134: (#H,"@121a8@26")
135: (#J,"q6")
136: (#K,"q8")
137:
138: (t5) @26@v102o3 BKZ @p90
139: (t5) r1 r*3456
140: (t5) [e1]
141: (t5) [e11][e12][e13][e14]
142: (t5) [e12][e13][e15][e16][e17][e18]
143: (t5) [e19][e20][e21][e23]
144: (t5)      [e20][e21][e23]
145: (t5) [e20][e20] ¥10[e20][e20]
146:
147: *e1* z127,107,107 r8{a<ce}8 FJ ab<c*12 >bga*12 deg*12
148:      HK {age}8{ged}8{d+dc}8 J dc>a*12 bge*12
149:      KB f8 F {fa<c}8 JG gfe*12 J f>a<c*12 de>a*12<
150:      KHG {gfe}8 J def*12 efd*12 gb<c*12>
151:      KHF X {ab<c}{ec>a}<{d>bg}a F {ged}{d+dc}{dc>a}
152:      {bgr}a BJ a*18b*18<c*12 K e>{a<c<c}>b<{d>bg}
153:      JZ >fa<c*12 gde*12 acd*12 ec>a*12
154:      KHG {b<cd}8 >g<gf*12 ecd*12 {gag}8b8
155:
156: *e11* XB
157: *e12* _10@d2 |:{rr'e<c'}:|'db'{'db'r'>b<g'}
158:       {rr'>b<g'}'ca' ¯10 H {ab<c}
159: *e13* e4 H >>{a<c<c} >b4<d*18>b*18g*12_20 E
160: *e14* |: 'ca''ca'|:3{rr'>b<g'}:|'ca''ca''>b<g' :|¯20
161: *e15* 'ca''ca'|:3{rr'>b<g'}:|'ca''>b<g''ca'
162: *e16* 'db''db'|:3{rr'ca'}:|'db''e<c''db'
163: *e17* r1r1¯12{age}{ged}{edc}{d+dc}
164: *e18* JZ >ag+g*12 d+dc*12
165:
166: *e19* K @d3 X _12
167: *e21* |: >a<e<c>ebege :|
168: *e22* |: >f<cacgcac :|
169: *e23* >f<cacgcac >g<dbdadbd
170: *e20* [e21][e22]
171: **
172: /---------- g.2
173:
174: (#H,"@121a8@25")
175:
176: (t6) @25@v84o3 BKZ @p90 @d3
177: (t6) r1 r*3456
178: (t6) [e1]
179: (t6) [e11][e12][e13][e14]
180: (t6) [e12][e13][e15][e16][e17][e18]
181: (t6) [e19][e20][e21][e23]
182: (t6)      [e20][e21][e23]
183: (t6) [e31][e20][e20] ¥10[e20][e20]
184:
185: *e31*  @26@v66 p1 @e120,24 r8.
186: **
187: ////////////////////
188: /---------- bd
189:
190: (t10) @17@v100@u127o2 L8
191: (t10) r2.c4
192: (t10) |:9 c{rrc}r{rrc}{rrc}crc :|
193: (t10) |: r{rrc}c :| rc
194: (t10) [f][f][f]
195: (t10) c4r2. c4r.cc16crc |:4 {rr}c16 :| {rc}{rc}
196: (t10) [f][f]
197: (t10) c1
198:
199: *f* |:8 c{rrc}r{rrc}{rrc}c|rc :| {rrc}r
200: **
201: /---------- sn
202:
203: (t11) @17@v100@u125o2 L4
204: (t11) r2.'fg'
205: (t11) |:9 @u125rdrd*16@u98d*32 :|
206: (t11) |: {drd}8r4 :| {dbg}8r8
207: (t11) [g][g][g]
208: (t11) r2.'fg' rdrd {ddr'd<d'}{'d<d'r'db''db'}
209: (t11) {r'da''da'r}{'dg'r'dg'r}
210: (t11) [g][g]
211: (t11) r1
212:
213: *g* |:8 @u125rd|rd*16@u98d*32 :| @u98{rrdddg} @u125d
214: **
215: /---------- hh
216:
217: (#M,"z110,80{frf}")
218:
219: (t12) @17@v100@u110o2 L8 k1
220: (t12) r1
221: (t12) [h1][h1][h2][h3]
222: (t12) [h][h][h][h5]
223: (t12) [h][h][h6]
224:
225: *h1* <c*0>|:32 M :|
226: *h2* <c*0>|:8  M :|
227: *h3* <@u110|:{crc}r4:|rc>
228: *h4* <c*0>|:28 M :| <c4 {rrc}r>
229: *h5* @u110<c*0>|:4 f4 :| <c*0 |:8 d :|>|:8 g :|
230: *h6* <c1>
231: *h* [h1][h4]
232: **
233: #ext_off
234:
235: (p)
```

リスト5 The End of Loveのカウンタ表示

```
 1:00003600 00000000   2:00003480 00000000   3:00003488 00000000   4:00003480 00000000
 5:00003480 00000000   6:000034A4 00000000  10:00002940 00000000  11:00002940 00000000
12:00002940 00000000
```

▶なんかむしょうに肉のカタマリが食べたいです。ああ，ビンボー学生はつらい。
青木　謙(19)神奈川県

◆キャミィのテーマ(スーパーストⅡ)

久々の「内蔵オンリーな正統派ゲームミュージックコピー」です。

音色がややイメージと違いましたが，原曲の雰囲気はよく出ています。楽譜を参考にした際にありがちな音の外れもないようです。ヘッドフォンで聴くと後半に突然パンが中央に固まってしまうので，チャンネルの使い方を改善してステレオで貫いてほしいところ。ドラムがちょっと奥にこもりがちなので，気になる人はEXTRA_PERCから選び直してみるのもいいでしょう。

近頃のゲームマシンに搭載される音源はPCMばかり(あのエンソニックの石も使われたりするしね)。内蔵じゃコピーは辛いし，巷ではMIDIが流行だし，いつの間にかこの手の作品も少なくなってしまいましたが，まだまだネタはたくさんあります。

内蔵音源はMIDIと同じくらい奥が深いので，ここから学べるサウンドのノウハウも測り知れないものがあります。まだまだ技術的にも開拓の余地があるかもしれません(私も新たな音を目指して鎌倉街道を爆走中)。皆さんも頑張ってください。

◆The end of love(オリジナル)

かっこいい曲です。テンションの高いバトルが，一定の展開を飽きのこないものにさせていて，なおかつこれが結構渋くて私好み。ソロのジャストなタメがグー。作曲者の鳥越さんもいっておられますが，まさにこれは生で演ってみたい曲ですね。

リバーブが強いかな？　ライブステージを意識したのでしょうか。SC-55の場合ちょっとモンヤリしてしまうきらいもあるので，レベルを少し下げてもいいですね。この曲の場合はキレもよくなると思います。ヘッドフォンのみの調節ということですが，やはり最終的にはスピーカーからの音を含め，いろいろな環境でチェックすることをお勧めします。

演奏について欲をいえば，もっと自然な流れが欲しいですね。ピアノやギターがちょっと硬く感じられます。抑揚感がしっかりしているだけに惜しい。

たとえば，ピアノの入りの部分は，

```
r＊1824 r @u76d＋＊2 @u127e＊23ga＊19
@u76b＊5@u127'e<c'47'd4b''>b4<g''>g<e''ca'
```

こんな感じで引っ掛かりをなくしたいところです。また，あちこちにQコマンドを入れると，さらによくなりますよ。

ところで，この曲はSC-55mkⅡ対応ですから，ノーマルなSC-55(CM-300，CM-500などを含む)の場合はリストをちょっと変更する必要があります。お手持ちの楽器がSC-55mkⅡ，もしくはSC-33の場合は問題ありません。

SC-55で演奏すると，液晶ディスプレイに「Data Error!」と表示されます。これは，ボイスリザーブの合計が24を超えるような設定が行われているためで(制作がSC-55mkⅡ上だから当然ですね)，これが同時使用可能ボイスが24のSC-55ではエラーとなって働きません。つまりSC-55にとって，リスト中のボイスリザーブ設定はまったく意味がないものとなります。

(進)の「ちょっといいですかぁ?」

で，SC-55対策として具体的にどうすればいいか，というと，まずリスト先頭部分にある，

```
.sc55vreserve $10=
{2,4,4,6,3,3,0,0,0,4,0,0,0,0,0,0}
```

という行，これをたとえば，

```
.sc55vreserve $10=
{1,4,4,6,3,2,0,0,0,4,0,0,0,0,0,0}
```

というように，総計が24以下になるように変更します。削ってもあまり影響はないと思われるパートを探して調節するのがいいでしょう。

次に，トラック4で使われているピアノ(バンク8の4番)ですが，SC-55ではこの音色が存在しません。代理発音機能でなんとかしのげるのでこのままでもいいですが，ホンキートンクは使用ボイス数が2と多く，曲のあちこちで演奏がブチブチ途切れてしまいます。これを防ぐには，使用ボイス数が少ない同系統の音色に置き換えてしまうのが確実です(といっても選択の幅は狭いが……)。

作者のあてた音色を変えるのは心苦しいものがありますが(いやホント)，音が切れるよりはマシでしょう。ダンパーを全部消すという方法もありますが，こちらはあまりオススメできませんからね(その前にコードパートをどうにかするという方法も，なくはないですが)。

そうそう，トラック1の先頭に，

```
y126,1
```

を加えてモノモードにしておくと，気持ち程度の発音数が稼げるかもしれません。音色にリリース音があろうと，モノ設定されたパートは常に1音しか使用しませんからね。まあ，スラップベースなのであまり問題はないでしょう。

これで，なんとかSC-55でも聴けるようになります(ああ，GS音源っていったい……)。

これらのことはいろんな状況で応用が利くものです。知らなかった人は頭の隅にでもとどめておいてください。パートをモノモードにするなんてのは，普段でも結構役に立ちますからね(SC-55のみならず)。

◆　◆　◆

さて，MIDI楽器の設定がイマイチわからずに苦しんでいるという読者の報告を多数受け，今回の増ページ企画となりました。

ここでは，楽器の個別設定も含めて，過去にも多かった質問への回答として「複数の楽器を同時に使うための設定」の一例を紹介します。ついでに「ARCCの応用，具体的な使い方」やら「SC-55のドラムキットや音色をいじくる方法」なんかもやっちゃいます(このへんは来月回しだけど)。

今回はDTM音源の標準的存在である，CM-64(MT-32)とSC-55を例にとりますが，基本的にはローランドエクスクルーシブでのセッティングを行いますので，ここを理解してもらえればほかの楽器(U-220など)への対応も楽でしょう(他メーカーの楽器を使用する場合は，エクスクルーシブメッセージについての知識が多少必要になりますが)。

今月は，CM-64とSC-55の同時演奏を実現するための，基本的な方法を解説します。

その前に，まず「トラック」「チャンネル」「パート」の違いを知っておくことが，自由自在なセッティングを行うために必要です。案外，ここで混乱してしまっている人が多いようです。

「トラック」は，もちろんZ-MUSICで用意されているトラックのことを指します。「チャンネル」とはMIDIチャンネル，「パート」は音源の編成単位です(たとえば，SC-55は16パートで1台の音源が構成されています)。

ここで，「パート」と「チャンネル」は別ものであることに注意してください。

たとえばデフォルト状態で「チャンネル10がドラムパート(なぜか普通はパート10)であっても(SC-55もそうです)，これは単に「チャンネル10がドラムパートにアサインされている」に過ぎません。楽器の設定によっては「チャンネル1でパート10を演奏」することもできますし，「パート1をドラムにしてチャンネル3で演奏」することもできます。「チャンネル1で16パート全部を演奏」というのも可能です。

さて本題に入りますが，普通に楽器をスルーでつないだ状態では，同じチャンネルで複数の楽器の「そのチャンネルがアサインされているパート」が同時に鳴ってしまいます。

これを防ぐには「楽器1のパート1は，チャンネル1で操作する」という旨の手続きを，すべての楽器のすべてのパートで行えばいいのです。いい換えれば，チャンネルは全部で16ありますから，「各チャンネルを，どの楽器のどのパートに割り振るか」を決めるということです。

いま，「全部で4チャンネルを使用してCM-64とSC-55を同時に演奏する」という状況を考えます(リスト参照)。

まずチャンネルをどう振り分けるかを決めましょう。チャンネル1～4を順に以下のパートへアサインするとします。

- ・CM-64　LA音源のパートを1使ってベース
- ・CM-64　PCM音源のパートを1使ってコード
- ・SC-55　パートを1使ってメロディ
- ・SC-55　パートを1使ってドラムス

ここまで決まればあとは簡単です。

リストのコメントでもわかると思いますが，ローランドエクスクルーシブ命令を使って，それぞれの楽器のパートごとにチャンネル割り当て情報を書き換えています。この場合，パラメータは0～16でチャンネル1～16，OFFが設定されます。使わないパートは，当然OFFにしておきます(パラメータは16ですね)。ほかの楽器をつないだ場合も，楽器マニュアルの「パラメータ・アドレス・マップ」や「パートの受信チャンネル変更」などの項目を調べて同様に設定してください。「?パートのマルチティンバ音源」ならば，そういった機能は備わっていることと思います(楽器によっては，パネルなどの手動切り

替えで対応しているものもあります)。

これがすめば，あとは普通にMMLを書いていくだけです。CM＆SCの同時演奏を考えていた人は，リストを入力して感触をつかんでみてください(曲は気にしないように)。なかなか簡単だと思いませんか？　ほかにもいろいろ方法はありますが，これがいちばんオーソドックスなやり方でしょう可能性も高く，管理が楽)。

◆　◆　◆

やっぱりDTM界主力のローランド音源に関しての解説は，いつかやらなくちゃならないことだったのね。

では，残りの細かな解説を残しつつ来月へ。

(進藤慶到)

リスト

```
  1: .comment 複数楽器同時演奏 && 基本設定例 && ARCC入門 by ENG
  2:
  3: / for ZMUSIC.X
  4: / MIDI MODULE : CM-64 & SC-55
  5:
  6: /----------------------------------------------------------
  7: / TRACK SETUP
  8:
  9: (i)
 10:
 11: .mt32_init
 12: .sc55_init
 13:
 14: (m1,1000)(aMidi1,1)               / CM-64
 15: (m2,1000)(aMidi2,2)
 16: (m3,1000)(aMidi3,3)               / SC-55
 17: (m4,1000)(aMidi4,4)
 18: (m5,1000)(aMidi4,5)
 19:
 20: /----------------------------------------------------------
 21: / CM-64 System SETUP
 22:
 23: / LA SOUND PART
 24: .roland_exclusive $10,$16 {$10,$00,$00
 25:          64                            / master tune
 26:           2, 4, 4                      / reverb (m,t,l)
 27: / Part#   1, 2, 3, 4, 5, 6, 7, 8, R
 28:          32, 0, 0, 0, 0, 0, 0, 0, 0    / ptl reserve
 29:           0,16,16,16,16,16,16,16,16    / MIDI ch#
 30:         100}                           / master volume
 31:
 32: / PCM SOUND PART
 33: .roland_exclusive $10,$16 {$52,$00,$00
 34:          64                            / master tune
 35:           2, 4, 4                      / reverb (m,t,l)
 36: / Part#   1, 2, 3, 4, 5, 6
 37:          31, 0, 0, 0, 0, 0             / ptl reserve
 38:           1,16,16,16,16,16             / MIDI ch#
 39:         100}                           / master volume
 40:
 41: / リバーブ，パーシャルリザーブ設定のほか，
 42: / LA パート1をMIDIチャンネル1，
 43: / PCMパート1をMIDIチャンネル2
 44: / へ割り当てている
 45: / マニュアルの「パラメータ・アドレス・マップ」と見比べて
 46: / きちんと理解すべし
 47:
 48: /----------------------------------------------------------
 49: / SC-55 System SETUP
 50:
 51: .roland_exclusive $10,$42 {$40,$00,$00
 52:         0,4,0,0}          / master tune
 53:
 54: .roland_exclusive $10,$42 {$40,$00,$04
 55:         127               / master volume
 56:          64               / master key shift
 57:          64}              / master pan
 58:
 59: .roland_exclusive $10,$42 {$40,$01,$10
 60:           0, 1,23, 0      / voice reserve
 61:           0, 0, 0, 0
 62:           0, 0, 0, 0
 63:           0, 0, 0, 0}
 64:
 65: .roland_exclusive $10,$42 {$40,$01,$31
 66:           5               / reverb character
 67:           0               / reverb pre-lpf
 68:         127               / reverb level
 69:          65               / reverb time
 70:           0               / reverb delay feedback
 71:           0}              / reverb send level to chorus
 72:
 73: .roland_exclusive $10,$42 {$40,$01,$38
 74:           4               / chorus macro
 75:           0               / chorus pre-lpf
 76:         127               / chorus level
 77:          30               / chorus feedback
 78:          65               / chorus delay
 79:           3               / chorus rate
 80:          25               / chorus depth
 81:           0}              / chorus send level to reverb
 82:
 83: / 以上，ごく普通の設定例
 84: / もちろん，SC-55専用命令を使える場所ではそれでも可
 85:
 86: .roland_exclusive $10,$42 {$40,$11,$02, 2}
 87: .roland_exclusive $10,$42 {$40,$12,$02, 3}
 88: .roland_exclusive $10,$42 {$40,$13,$02,16}
 89: .roland_exclusive $10,$42 {$40,$14,$02,16}
 90:
 91: / SC55のパート1,2をMIDIチャンネル3,4に割り当てている
 92: / このままではPART1,2とPART3,4での使用チャンネルが
 93: / かち合うので，PART3,4はoffしなくてはならない
 94: / 曲によって使用チャンネルやパート構成が異なるので
 95: / 各自設定する必要あり
 96:
 97: / なお，以上の設定は
 98: / .sc55_part_setup 1 {3}
 99: / .sc55_part_setup 2 {4}
100: / .sc55_part_setup 3 {17}
101: / .sc55_part_setup 4 {17}
102: / これらのコマンドを実行することと変わりはない
103:
104: / マニュアルの「パラメータ・アドレス・マップ」と見比べて
105: / きちんと理解すべし
106:
107: /----------------------------------------------------------
108: / MML DATA SET
109:
110: (t1)    @i$41,$10,$16 t86         / CM-64
111:         r8
112:         y126,1                    / MONO
113:         @32 @v110 p3
114:         x3,0,6,0·                 / REVERB OFF
115:         r4
116:         @u120 L16 o2 @q3
117:         |:32c:|
118:
119: (t2)    @i$41,$10,$16             / CM-64
120:         m,1                       / 拡張ARCCを使う
121:         s,2 @s,22 @h,3            / 波形、スピード、待時間
122:         r8
123:         @31 @v95 @p70
124:         r*47
125:         @c11,127,127              / EXPRESSION
126:         @a70                      / ARCC ON
127:         @u90 L16 o4 q8
128:         'ce-gb2''ce-gb-2''ce-ga2''ce-gb-2'
129:
130: (t3)    @i$41,$10,$42             / SC-55
131:         m,1                       / 拡張ARCCを使う
132:         s,0 @s,2 @h,3             / 波形、スピード、待時間
133:         r8
134:         y126,1                    / MONO
135:         @82 @v80 @p55
136:         @e70,20
137:         @y1,$20,64                / TVF ｶｯﾄｵﾌ･ﾌﾘｸｴﾝｼｰ
138:         @y1,$21,88                / TVF ﾚｿﾞﾅﾝｽ
139:         @y1,$63,58                / TVF&TVA ｱﾀｯｸ
140:         @y1,$66,60                / TVF&TVA ﾘﾘｰｽ
141:         y$63,$01                  / NRPN MSB
142:         y$62,$20                  / NRPN LSB
143:         r4
144:         @c6,64,70                 / DATA ENTRY MSB
145:         @a28                      / ARCC ON
146:         @u113 L16 o4 @q3
147:         g2f2e-2f2
148:
149: (t4)    @i$41,$10,$42             / SC-55
150:         x$40,$12,$14,1,1          / LIMITED-MULTI ,
151:                                   / PART2 = RHYTHM (MAP1)
152:         r8
153:         @33 @v118 p3
154:         @e70,12
155:         @y24,35, 63               / ﾄﾞﾗﾑ･ﾋﾟｯﾁ
156:         @y24,40, 69
157:         @y26,35,127               / TVA ﾚﾍﾞﾙ
158:         @y26,40,122
159:         @y29,35, 10               / ﾘﾊﾞｰﾌﾞ･ｾﾝﾄﾞ･ﾚﾍﾞﾙ
160:         @y29,40,120
161:         r4
162:         L16 o2 q2
163:         z127,120,127,120, 127,110,127,110
164:         |:4c-rer c-c-er:|
165: /       ベロシティーシーケンスは，コンパイル時に
166: /       マクロ展開される
167: /       常にMMLと対をなしていることに注意
168:
169: (t5)    @i$41,$10,$42             / SC-55
170:         r8
171:         @y26,42, 95               / TVA ﾚﾍﾞﾙ
172:         @y26,46,105
173:         @y28,42, 20               / ﾊﾟﾝﾎﾟｯﾄ
174:         @y28,46, 20
175:         @y29,42, 60               / ﾘﾊﾞｰﾌﾞ･ｾﾝﾄﾞ･ﾚﾍﾞﾙ
176:         @y29,46, 60
177:         r4
178:         L16 o2 q2
179:         @u96 <a*0>
180:         z127,110,120,110, 127,110,127,120,
181:         |:4f+f+f+f+ f+f+f+a+:|
182:
183: (p)
```

SIDE A

ドライビングモード詳説

Tan Akihiko 丹 明彦

今回はプログラムを高速化するため，多段階ディテールの導入を考える
実際に何割か実行速度を稼ぐことができ，かなり操作性を向上させることができた
さらに，サーキットのソーティングについても触れておく

デイトナUSAとバーチャレーシングMD，ジオグラフシール

普通免許を手にしたとたんに教習所シミュレータを作る気もあっさりと失せてしまったが，また例によって近頃目についたものをピックアップしてみよう。

大物はセガの「デイトナUSA」。「リッジレーサー」の向こうを張るようなテクスチャマッピングばしばしの3Dレーシングゲームだ。となればどうしても「リッジレーサー」と比べてしまうわけだが，目新しさという点では「リッジレーサー」を最初に見たときの戦慄にはほど遠いのはしかたない。2Dものや疑似3Dものとたいして変わらない画面の質感も，セガの主張なのだろうが個人的には好きではない。それでもゲームとしての仕上げ，充実した演出はセガの強みであろう。サーキットはさすがに凝っている。パイロンや道路わきの銅像など遊びの要素も多い。

試しに数回遊んでみたのだが，ドライビング感覚はまだよくつかめない。特にコーナリングの感覚が，強引に車を曲げてカウンターを当てまくる「リッジレーサー」と結構違う。コースをある程度覚えて早めに対処しないと曲がっていけないので，アンダーステアが強いように感じるのだ。

が，きちんと走っている人のプレイを見ていると気持ちよさそうだ。初見で魂を奪われなかった，という点ですでにいまひとつなのではないかという懸念があったのだが，走り込んでいくうちに少しずつ楽しくなってきた。それはそうと，スタート時の「GENTLEMEN, START YOUR ENGINES」のメッセージは女性プレイヤーに対して失礼だぜ。かのインディ500でも，女性ドライバーが出場すれば「LADIES AND GENTLEMEN……」とやってくれるのだ。あのまま米国あたりに輸出すると訴えられたりしてね。

それから，これまたセガの新作だがメガドライブの「バーチャレーシングMD」。なかなかに天晴れな出来栄えだといえよう。ポリゴンモデルを簡略化してでも操作性を保とうとする姿勢は正しい。カートリッジ内にポリゴン専用のチップを積んでいるという話だが，あの値段であのパフォーマンスはなかなか。秒間9000ポリゴンだそうで，毎秒15フレームとしても1フレームあたり600ポリゴンか。実はこれはSLASHをX68030で動かして到達できないレベルではないが，相当難しいことも確か。その難しさはSLASHの性能というよりも制御するプログラム側にある。メガドラのプログラマやデザイナーはきっと死ぬ思いをしているのだろう。

プログラマやデザイナーが死んでるんじゃないかといえば，エグザクトの最新作，「ジオグラフシール」も結構キてる。どうしてあの人数，あの開発期間であれだけのものができてくるのか，まったくもって驚異だ。

ドライビングモードの解説

今回は先月のコースエディタのうち，ドライビングモードを解説する。また最近，ちょっとした改造で高速化し，操作感覚を向上させたのでその手法も解説しておく。ちょっとした改造といっても，ソースプログラムの差分はそれなりに大きい。テスト実行時間も合わせて数時間で行った仕事ではあるけれども，遺憾ながら誌面に掲載できる量ではない。細かいバグフィックスもいくつか行っている。

プログラムの流れ

とりあえず全体を眺めてみる。

1) コースを作る

エディタモードまたはファイルを読み込む。今回は詳しい解説を省略する。

2) ドライビングモードの準備をする

スーパーバイザモードに入り，画面モードやマウスなどの初期化やSLASHの描画モードの設定を行う。車をスタート地点に置く。現在時刻を記憶する。

3) メインループ

実際にコースの上で車を走らせる。

4) 後始末をして終了する

画面モードやマウスの動作モードを元に戻し，ユーザーモードに戻る。

それでは3)のメインループを詳しく見てみる。5月号の付録ディスクに収録されているdrive.cのdrive()関数を参照すると，なお理解が深まることだろう。

(A) マウスの入力を得る。これはハンドル，アクセル，ブレーキへの操作とみなされる。

(B) キー入力を得る。これは終了やエディタモードへの切り替え，グラフィックVRAM描画モード/テキストVRAM描画モードの切り替えの操作なので適宜処理を行う。なお描画モード切り替えの場合はパフォーマンスモニタによる平均fps値(frame per second，1秒間に何コマ描き換えを行ったかという値）を標準出力に出力する。プログラム実行時にリダイレクトでこれを受けておくと，どのくらい滑らかに動いているかの指標として記録に残すことができる。行頭にdisplaymode=0と書いてあるのはテキストVRAM描画モードのfps値，displaymode=1と書いてあるのはグラフィックVRAM描画モードのfps値である。

(C) 現在の車速，車体の姿勢と(A)で得たハンドル，アクセル，ブレーキ，および直前のフレームからの時間差(現在時刻から割り出す)から次の瞬間の車速と暫定の姿勢を得る。関数driveMain()がこの処理に当たる。なおこの処理は4月号に説明した「斜面までサポートした車の運動」とまったく同じである。

(D) 接触判定によって暫定の姿勢に必要なら修正(別の斜面に乗った場合はその斜面に沿わせ，壁にぶつかった場合は動きを止める）を加え，次の瞬間の姿勢を得る。関数checkcollision()がこの処理に当たる。この処理も4月号の手法と基本的に同じ。異なるのは次の3点。コースを複数のブロックで構成したことから，各ブロックの平行移動量を考慮に入れているということ。同じく複数のブロックという理由から隣のブロックに移るための処理を追加したということ。また立体交差が同一ブロックの中には存在しないという理由から，接触判定の検索起点を車体のかなり上方にとって車体が路面にめりこむ危険性を減らしたこと。このプログラムでは複数のブロックによってマップシステムもどきを実現しているので，のちほどまとめて解説したい。

(E) コースを表示する。この処理は関数display()に当たる。

(E1) 描画するページ(ダブルバッファのどちらに描くか）を指定する。ダブルバッファリングは同様の画面を2ページもち，表示していないページに描画を行い，描画が完了したらページを切り替えることでちらつきのない画面描き換えを実現することができる手法である。

(E2) 車の姿勢から視線を求め，視線から変換行列を求める。そして，光源計算を行う(SLASHライブラリのGetRayView() 関数を用いている)。

(E3) コースを構成するブロックのうち表示するブロックをピックアップする(関数findDisplayBlock())。視線の方向の単位ベクトルと視点からブロック位置方向の単位ベクトルとで内積を取り，ある一定値以上ならば表示する。なお例外として，自車が乗っているブロックは必ず表示する。単純に内積のみで判定するとそのブロックが表示されない可能性があり，具合が悪いためである。

(E4) 各ブロックを視点からの距離でソートする(関数findDisplayBlock())。先月の時点では「まだプログラムしていない」と書いたが，付録ディスク収録のソースプログラムにはこの処理が入っている。連載の締め切りと付録ディスクのマスターアップの時期の微妙なずれから生じた不整合ということで，実害はないのでご容赦いただきたい。

(E5) 変換行列(車の姿勢を表す基底ベクトルを並べて作る)と平行移動成分(ブロックの位置と視点の位置との相対ベクトル）によって透視変換を行い(SLASHライブラリのTranslateByMatrixView()関数を用いている)，描画を行う(SLASHライブラリのDisplayPolygonList() 関数を用いている)。(E4) のソート結果を利用して遠くのブロックから描画する。視点からある程度遠いブロックに関してはディテールを落としたモデルを表示する。なお，

先人の仕事に見る「快適なfps値とは？」

いまごろになってメガドライブを購入した。買ったソフトは「バーチャレーシング」のみ，実にわかりやすい。というわけで「バーチャレーシング」購入を記念して，現在私が目標としている3Dポリゴン系カーレースゲームのfps値を計測してみた。fps値の計測は，いったんビデオに録画して，30フレームをコマ送りする間にゲーム画面が何回描き変わったかを数える方式で行った。

・「Indianapolis 500」: the simulation

マシンの速度によってfps値が変動するが，AMIGA A 1200では12〜15 fps。十分に滑らかで運転しやすい。上空から眺めるモードでは7〜12 fpsに低下する。

・「World Circuit」(Formula One Grand Prix)

変動するがAMIGA A 1200では7〜9 fps。動きの粗さを感じる場面もあるが，ステアリングのアシストなどのおかげでまあ運転できる。

・「バーチャレーシングMD」

常に15 fps。視点をどう変えてもそう。ゲームメーカーの力量といったところか。ポリゴンの出現消滅を見ているだけでも参考になる。

結局，15 fpsは快適，12 fpsで満足，8 fpsは最低線といえる。

この2段階ディテールの部分は先月号の付録ディスクには入っていない。
(F) パフォーマンスモニタを更新する。
(G) ダブルバッファリングを行ってページを切り替え，描画の完了したページを表示し，メインループの先頭に戻る。

ランタイムにおけるソート

さて，本プログラムで新しく取り入れた手法について解説を加えていくことにしよう。

まずはソートである。上の解説でもわかるとおり，このプログラムではサーキットを構成するブロックを視点からの距離によってソートし，遠くから描画している。これにより，ブロック単位での前後関係が保たれ，おおむね破綻しない表示が得られている。これをさぼると，壁の向こうに遠くの道が見えてしまうなどといった現象が起きてしまうのだ。視点の位置は毎フレーム変化するので，視点からの距離によるソートも毎フレーム行う。

実は，ポリゴンシステムとソートの関係について語ることによって，このようなデザインに到達した背景をより深く理解することができるのだ。

まず，3次元のポリゴン表示システムにおいては，物体の前後関係が破綻しない，つまり近くの物体は常に遠くの物体に優先して表示されることが要求される。そしてこの要求を満たすことはそれほどやさしいことではない。世の実用的な多くのシステムは，表示を破綻させないためにデプスソートやZバッファなどを用いている。

デプスソートとは，透視変換後の各ポリゴンを画面に向かって奥行き(デプス)順にソートしたのちに描画する手法である。ところがデプスソートは不用意に使えない。デプスソートは，ポリゴンの枚数が多くなるほど処理が重くなるという宿命を背負っているのだ。ある程度見映えのする世界を作ろうとすると，どうしても数千から数万ポリゴンは必要になる。それを正直にソートすると，ソートのための計算量が馬鹿にならない。蛇足だが，ポリゴンシステムの開発者はそれほど正直ではいけない。たとえば，直前のフレームでのソート結果を最大限に利用して2回目以降のソート処理を軽くするなどの手段は講じておきたいものだ。

対してSLASHは，ポリゴン単位でのデプスソートを行わないポリゴンシステムである。とにかくポリゴンリストに定義した順番でポリゴンを描くという極めてシンプルな姿勢で設計されている。むろん表示が破綻する可能性は大きいのだが，それはモデリング段階で抑える方針になっている。事実，意外に多くの形状が，モデリング段階でのソートだけでおおむね正しく表示することができる。これに気づいている人は実のところ少ないようである。非力なマシンの上でできるだけ速度を稼ぎたいSLASHとしては，この方式を採用するのは必然の流れなのである。なお，SLASHアプリケーションは「おおむね表示がうまくいっていればそれでよし」という潔い態度で制作されるということも付け加えておこう。

念のためいっておくが，「表示時の」ソートが重いと嫌われるのであって，「モデリング時の」ソートは重くても嫌われないのである。ついでにいっておくが，モデリング時のソートは現在のところ人間の仕事である。一応ソート関数SortPoly()は用意してあるのだが，ソートが破綻するポリゴンリストについてはソートは失敗する。その破綻が通常はめったに起こらないような小さな破綻であっても，SortPoly()は「破綻が大きいか小さいかという評価を行ってできるだけよく見えるようにソートを行う」という処理をしないので，ソート結果がぐちゃぐちゃになってしまう。できるだけ破綻の少ない順番にポリゴンを並べ替えるのは経験を積んだデザイナーの力量に頼っているというのが現状である。

さらに，優秀なデザイナーをもってしてもソートできない形状も存在する。そのときにはポリゴンマクロといって，3つまでのポリゴンリストを表示時にポリゴンリスト単位でソートする機構を使うことになる。これは100枚単位，1000枚単位のソートともなると一瞬ではすまないが，3要素までのソートなら一瞬ですむという理由から導入を決定したものである。SLASHの開発は，常に効果と処理量のバランスを取りながら行われているのだ。

そして今回のブロック単位のソートである。モデリングの段階で吸収できるのはせいぜい1ブロックの中でのみ。シーン全体に存在するポリゴンは数千ポリゴンに達するのでデプスソートは問題外。だがブロック数はせいぜい数十個だから，ソートしてもどうにか問題なくすみそうだ。ということで現在のデザインに決定したのである。ちなみにソートのアルゴリズムにはシェルソートを用いている。ここは改良の余地もある。どうせブロック単位なのだから各ブロックについてあらかじめソートした結果を保存しておき，自車のいるブロック番号をキーとするテーブルを作るのが効率的であろう。

多段階ディテール

これが今回の改良の目玉。高速化と操作性の向上に大きく貢献している。ここでも多段階ディテールの必要性について語っておこう。

3次元コンピュータグラフィックとは，遠近感によってものを立体的に見せる手法といっていい。つ

まり，遠くにある物体ほど小さく見える。もう少しいうなら，遠くにある物体ほどたくさん表示しなくてはならない(同じ画面上の面積なら,表示の小さいほうがたくさんつめこめると考えればわかりやすい)。遠くの物体は,表示が小さくて潰れるわりには,透視変換にかかる計算量は同じだし，描画面積が狭いからといってそれほど処理が軽くなるわけでもない。これは損である。

そこで，ひとつの物体に対してそれを表現するための複雑さの異なる形状をいくつか用意しておき，その物体が遠くにある場合は簡単な形状(低ディテール)を，近くにある場合は複雑な形状(高ディテール)を用いる。これにより，遠くの物体を表示するための処理が極端に重くならなくてすむ。これが多段階ディテールの考え方である。

今回のプログラムでは，効果を見る意味でとりあえず2段階ディテールを導入した。具体的には低ディテールの形状を追加した。前回説明したコース部品は高ディテールとして用いる。低ディテールのコース部品は，スケルトン(骨組み，5月号参照)は共通でもポリゴン構成は大幅に簡略化してある。具体的には，センターラインがない，縁石がない，極めつけは進行方向の分割をなくした。高いディテールのコース部品では4(=PC_DIV) 分割されていたので，結果として頂点数とポリゴン数を大幅に減らすことができた(図1)。

2種類の形状の選択は，視点からブロックまでの距離がある定数(しきい値)を超えたか超えないかで行っている。

2段階ディテール導入の効果を数字で見てみる。比較対象は2段階ディテール導入前のシステム，2段階ディテールを導入したシステムに関してはディテール切り替えのしきい値を変えてみた。それぞれのシステムで，先月の付録ディスクに載せたサンプルのコースを周回してみて平均fps値を計測してみた。

以下はその結果である。計測はX 68030，キャッシュを有効にした状態で行った。

[1] 2段階ディテール導入前(高ディテールのみ)
　テキストVRAM描画モード　7.0 fps
　グラフィックVRAM描画モード　5.6 fps
[2] 2段階ディテール，しきい値=ブロック長×5
　テキストVRAM描画モード　8.8 fps
　グラフィックVRAM描画モード　6.6 fps
[3] 2段階ディテール，しきい値=ブロック長×2
　テキストVRAM描画モード　9.6 fps
　グラフィックVRAM描画モード　7.2 fps
[4] 参考(低ディテールのみ)
　テキストVRAM描画モード　11.0 fps
　グラフィックVRAM描画モード　7.6 fps

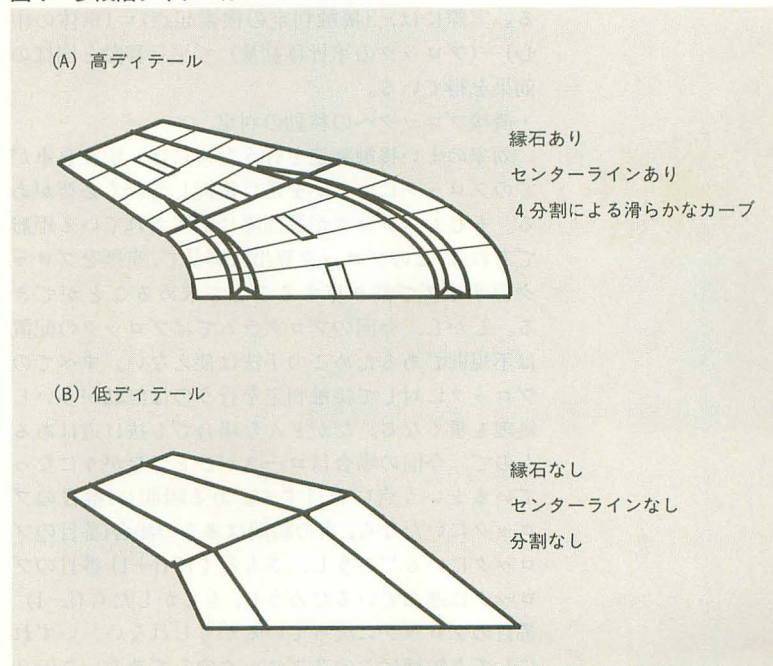
数値で見ると劇的に速度が上がっているようには見えないが(といっても[1]から[3]で約40%の高速化である),実際に操作してみると数字以上に効果が上がっている。特に,サンプルのコースでは1コーナーを立ち上がった頃に表示するブロックが一気に増えるのでがっくりとスピードが落ちていたのだが，2段階ディテールを導入しただけでほとんどもたつかなくなったのである。経験的には，5 fpsを下回ると操作性が極端に悪化するのだが(まあこのへんに関しては，ハンドルがクイックすぎてほとんど私にしか運転できないあのプログラムも悪いといえば悪い),多段階ディテールの導入によってクリアできそうである。

なお，コース描画に関してはテキストVRAM描画モードがグラフィックVRAM描画モードに比べて有利であるということも上の結果から読み取れる。

いい加減な形状を使うわけで，画質の劣化が心配になるところだが，心配したほどのことはなく，むしろすっきりとしたくらいである。近くの路面は高ディテールで描いているので，路面の情報もきちんと見て取れる。たった2ブロック先までしか高ディテールで描かないように設定してもほとんど気にならない。

試しに[4]の低ディテールのみというコースで走ってみたが，これはさすがに粗かった。コーナーがきちんとカーブを描いていないというのはやはり気持ちの悪いものである。今後カーレースシミュレータが完成に近づけば，このへんのバランスの取り方が重要になっていくのだろう。

図1　多段階ディテール

ハードコア3Dエクスタシー(第9回)

マップシステムもどき

本プログラムで使用しているコースは，カーブまたは直線の先に次のブロックを継ぎ足していく方式で構成しているため，マップシステムを利用することができない。基本的にマップシステムは等間隔の矩形ブロックで構成されたシーンに対して用いるからである。そこで本プログラムでは，ブロックを自前で管理することにした。全部Cで書いたし，またブロック数が少ないうちはそれほど重くもならないだろうと考えて最適化を施していないため，まだかなりの無駄がある。私のプログラミングスタイルはいつもこうで，とりあえず動くことが大事，あとから速くしていけばよいという姿勢でやっている。個人的な事情はさておき，このマップシステムもどきを実現するにあたってはいくつかの特徴的な処理があるので列挙する。説明ずみのものもあるが，マップシステムという観点から再度解説しておく。

・ブロックの表現

ブロックを構成するポリゴンの頂点の座標は，その頂点が実際に存在するワールド座標でなく物体座標で記述し，表示する時点で平行移動している。これは，SLASHの制約から，安全に扱える頂点座標の範囲が限られているためである。

・平行移動成分を考慮した接触判定

車の基底座標系(位置と姿勢)はワールド座標である。ブロックと，ブロックに付属している接触判定用の情報は物体座標系である。接触判定を行うときはこの差を埋める必要がある。それが平行移動である。実際には，(接触判定の検索起点)=(車体の中心)−(ブロックの平行移動量)で平行移動と同様の効果を得ている。

・隣接ブロックへの移動の判定

効率のよい接触判定を行うためには，現在自車がどのブロックにいるかを常に把握しておく必要がある。もしもブロックが等間隔に配置されている矩形であれば，このブロック算出は容易で，座標をブロックのサイズで割り算することで求めることができる。しかし，今回のプログラムではブロックの配置は不規則であるためこの手法は使えない。すべてのブロックに対して接触判定を行うのは無駄が多いし処理も重くなる。だがどんな場合でも抜け道はあるもので，今回の場合はコースがひとつながりになっているという点に着目する。ある瞬間にi番目のブロックにいたなら，次の瞬間は多くの場合i番目のブロックにいるだろうし，さもなくば(i+1)番目のブロックに進んでいるだろうし，もしかしたら(i−1)番目のブロックに戻っているかもしれない。いずれにしても候補はこの3ブロックのみである。この3ブロックに対して接触判定を行えば十分ということになる。むろん，サーキットはループしているから，最初と最後のブロックでは例外処理が必要である。また，分岐をサポートするようになれば，もう少し複雑なデータ構造を導入する必要も出てくるだろう。

・視点からブロックへの相対ベクトルの算出

ここからは表示時の処理である。共有できる情報が多い。(相対ベクトル)=(ブロックの平行移動量)−(視点の座標)である。

・視点からブロックへの距離の算出

これは前項の相対ベクトルの大きさ。ついでに相対ベクトルをこの大きさで割れば，相対ベクトル方向の単位ベクトルになる。

・簡易クリッピングによる不要ブロックの間引き

視線の単位ベクトル(基底座標系のγ軸)と前項の単位ベクトルとで内積を求め，あるしきい値を下回ったらそのブロックは表示されないものとして一切の処理を省略する。たとえばしきい値に1/2を用いたら，$\cos 60°=1/2$より120°の視野に入らないブロックを間引くことになる。なお，自車のいるブロックは無条件で表示する。

・ブロック単位のソート

上で求めたブロックへの距離を用いてソートする。詳細はすでに述べた。

・多段階(2段階)ディテール

上で求めたブロックへの距離を用いてディテールを切り替える。詳細はすでに述べた。

今後の活動についての提案

最近SLASHのCライブラリのリファレンスマニュアルを書いた。が，LaTeXで書いたということと，量が多すぎるということで，掲載することができない。内容的には先月の付録ディスクに収めたドキュメントファイルと同じなのだが，やはり紙にマニュアルが印刷してあると読みやすさが圧倒的に違う(UNIXのX-WindowやOS-9のようなマルチウィンドウ環境だと，オンラインマニュアルも使いでがあるのだが)。SLASH自身も巨大すぎて付録ディスクのお荷物になっている感があるし。

そこで，特にSLASHに関心のある方々(マニュアルがちゃんとしてたら使おう，という人はいるだろう)を対象にしたサポート形式を検討している。つまり，リファレンスマニュアルやSLASHの最新版の回覧を行ってバグレポートやバージョンアップを行う。媒体の候補としては，郵便かパソコン通信を考えている。3Dポリゴンをぶんぶん振り回してなにか凄いものを作りたいという野望をお待ち申し上げる。詳しいことは来月号にて，ではごきげんよう。

SIDE B

SIDE B
マップシステムの制作

Yokouchi Takeshi 横内 威至

高速化するために，やらなくてすむようなことはなるべく避けたい
そこでマップシステムによって，空間内の情報を把握し，表示の無駄を省くことを考える
用途によって作り変えなければならない煩雑さはあるが，非常に有効な手段だ

SLASH ver.2.0のリリースも終わり，いろいろなサンプルも公開できた。今回は，空間での制御に向けてマップシステムを考えてみようと思う。すでに丹氏が解説した部分でもあるが，ひととおりコーディングレベルまでを考えることにしよう。コーディングすることによって，結構盲点が多いことに気づく。ということでいろいろな問題点も考えたい。

マップシステム自体，本来SLASH ver.2.0に載せる可能性もあったのだが，用途によって仕様がいくらでも変わってしまい，汎用性に欠く領域となっているので，結局載せないことになった。だが，3Dモノとしては基本となる部分であるため，3Dでなにかをしようと思うのならばこのマップシステムが必要となるであろう。したがって，今回は基本的な内容について固めておく。

その前に，オイラー角の領域でお便りをいただいた。ZENZOさん，どうもありがとう。あと，公開したSLASH ver.2.0にバグがいきなり発見されてしまった。申しわけない。バグは，スクリーンサイズ設定の部分で，引数として与えるY方向のドット数を2の倍数に限定してもらいたい。そうしないと場合によっては，画面をクリアしたときに下1ドット残ってしまうのことがあるだ。

マップシステムの効果を考える

まずは，3D空間内で自由に動き回ることを漠然と考えてみる。すると，空間中にあるオブジェクトすべてを調べて表示するのが最も簡単な方法であることに気づく。

しかし，当然のことながら画面上に現れない物体が多数存在する。というよりもむしろ画面に表示される物体は全体のわずかな部分でしかない。では表示されない物体さえも座標変換し，画面に入るかどうかをマメに調べなければならないのだろうか。これは，明らかに無駄な作業である。ということで，画面にそれほど影響を及ぼさないオブジェクトを省いて表示につなげるのがマップシステムの役割なのだが，どのオブジェクトが「影響を及ぼさない」のかを考えてみなければならない。

ここで，マップシステムが組み込まれた一例を思い出してみよう。アルシスの名作「スタークルーザー」の地上面である（知らない人はごめん）。要するに3Dの迷路場面である。このゲームでは，自機を中心としたマップが表示されていたのだが，これを見ればどんなシステムであるかは容易に考えられるだろう。マップを絶対座標に固定し，自分はその座標系内を移動する。マップはちょうどX68000のスプライトBGのようにピクセルとして扱い，自分の位置のピクセル周辺のみを表示しているだけだ。

しかし，これでも十分にマップシステムといえる。それに無駄を省く，という意味では十分効果を上げている。ところが，この方法だけでは3Dの効果としてはもの足りない。なぜなら空間を把握するのに十分な情報が不足していたからである。どういうことかというと，単純に遠くが見渡せないのである。結果的に自分の位置をつかみにくく，空間を把握できずに迷ってしまう。これは遠くが見えないという理由だけではないのだが，3Dの醍醐味であるリアルな，あるいは過激な遠近感を損なっているのだ。迷路内であれば，あまり必要ないことではあるのだが。

さらに，偉大な3Dゲームであった「スターブレード」も考えてみよう。これは無駄を省くというようなセコイ考えは捨てていたかもしれない。ひそかな売りとして「無限遠方」までをスキャンしていたような覚えがある。マップシステムらしきものがプログラムされていたとしても，あくまでも広大な空間を管理するためのシステムでしかなかったはずだ。

3Dであれば可能なかぎり遠くまで見えたほうがいいに決まっている。しかし，遠くになるほど物体は小さくなる。さらに遠くを投影するには，かぎりなく広大な空間を表示しなければならないので，マ

シンの負担が極端に増えることになってしまう。「スターブレード」はかなりのレベルまでやっていたと思うが，ごまかしもうまかった。惑星上でのシーンで，どこまでも見渡せるようなシーンを作らなかったのは，おそらくごまかすためもあったのだろう。

最後にもうひとつ，「バーチャレーシング」である。かなり細かいコースであったが，当然コース全体を一気に表示してはいない。かなり遠くまで見渡せはしたが，注意して見ると遠くの地形がある瞬間突然現れているのに気づく（バーチャモード4で見るとよくわかる）。当然マップシステムを導入しているのだが，さすがはハードレベルのポリゴン処理。1つのピクセル(というよりブロックとしておこう)はかなり巨大であり，遠くでの表示だからといって地形のディテールを一切落としたりしていない。

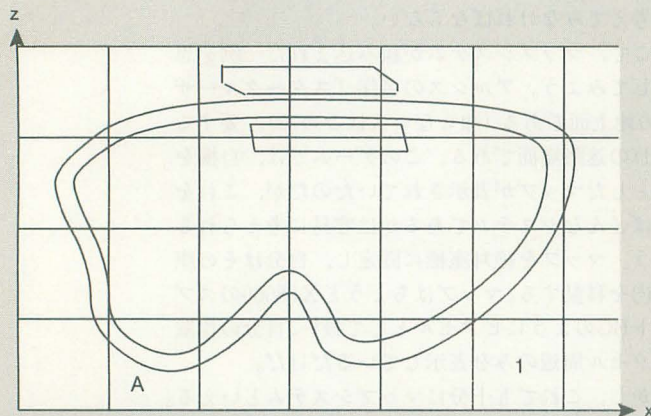
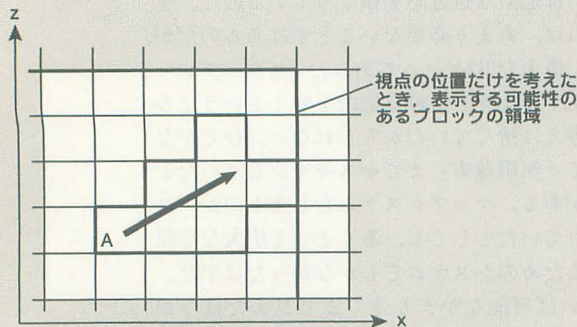
図1　マップシステムの概要

こんな感じで等分割する。各ブロックがちょうどスプライトBGのチップのように作られる
分割するサイズは，作りたいものによって都合よく設定するのが望ましい。たとえば，図中の1の部分では，分割サイズによっては道路が引っかかってしまい，空きブロックにならない。わずかでも道路が入ってしまうと無駄にデータを増やすことになる
実際には，動作の細かさ（描画の速度ではない）によって，データ上の分割の長さを決定していく。さらに，コースデータを作りながら，どこまで見渡せるか，どの程度細かい描写をするかなど，さまざまな要素を考える必要がある

図2　表示範囲

Aに視点があり，矢印方向を見ているとき，図の太線に囲まれたブロックを表示する。そして，このような表示ブロックを視線の方向によって何段階か作っておく

本当なら別のタイプのもの「エアーコンバット」も考えなければならないのだが，現物を見たことがほとんどないのでパス。フライトシミュレータならば無限遠方までを真剣に考えなければならないのだが，先ほどのようにかなりの問題を含むのでドライブものとは別で扱うべきであろう。

マップシステムの機能

ということでマップシステムにどんな機能を導入するかを考えよう。以上のような例を考えると，3Dらしさを失わないために，より効率を上げるにはどうすればいいかが浮かんでくる。

基本的には空間を管理するためのシステムであるが，効率を追究，つまり処理速度の向上を目的とする。また，より遠くまでを見渡せることも重要なのだが，遠くの物体はかなり小さく表示されることになる。となれば遠くの物体は細かい描写をする必要はない。どうせポリゴンがつぶれてしっかりと認識することができなくなるからである。よって，遠くに見える物体は簡略化した物体に置き換えるようにするのも効率向上につながる。では具体的な効率向上の方法を列挙しよう。

1)　絶対座標系をブロック化し，適当な配列によって任意のブロックをポイントできるようにする
2)　視点の位置によって表示すべきブロックを適当に選択する
3)　表示すべきブロックと視点との相対距離によって表示するディテールを変化させる

以上がマップシステムの大体の役割となる。さらに深く追究すると空気遠近法なんかも絡むし，当たり判定も含まれることになるであろう。

具体的な仕様

まず用途は普通にありそうなもの，地上があってその上を動くためのものとする。さらに，これから作ろうとしているのはそんなに遠くが見える必要がないと仮定しておこう。いきなりフライトシミュレータのためのマップシステムを解説するのも無謀なことなので，本当に基本的なものを作ってみた。もちろん応用すればなんにでも使えるだろうが，今回は基本的な話なのでとりあえずのレベルにする。

それではまずブロック化であるが，図1に示そう。きわめて単純であるが，結局このような形が最も扱いやすいのではないだろうか。長方形内を矩形に区切っただけという，かぎりなく基本的なものだ。このような方式は正にスプライトBG画面と同じである。実際，各ブロックのオブジェクトをポイントする動作はスプライトBGの動作とかなり似ている。そのあ

たりはリストの解説と一緒に考えることにしよう。

さて，それではどのブロックが表示すべきブロックなのかを調べなければならない。単純に考えると視点が存在するブロックの周り，いくつかのブロックだけを表示することになる。しかし，それでも当然無駄がある。となれば，そのブロックのうちの視線方向周辺のブロックを表示するのが賢いだろう。図2に簡単に示す。今回のリストでは方向をオイラー角から得ているが，やはりしっかりと行列から求めるべきである。まあそんなに複雑なものではないのでわかりやすくオイラー角で解説していこう。

コーディングする

今度は具体的なアルゴリズムである(図3)。実際のマップデータだが，これは横方向にX軸，縦方向にZ軸をとる。これはシステムの都合なので理解しにくいかもしれないが，がまんしてもらいたい。

そして，データとして登録するのは各ブロックの頂点リストのアドレス，ポリゴンリストのアドレスとなっている。マップデータと呼んでいるが，実際はポインタのようなものである。

まず，サンプルではブロックの1辺を2048，全ブロック数を64×64ブロックとしてある。必要とあればこの値を適当に変えてもかまわない。で，これらのブロックを視線によってオフセットして表示する。地上と考えず，整然と並んだ物体と考えたほうがわかりやすいかもしれない。

以上のようなものを最終的に表示するのだが，このマップデータのうちどれを表示するのかがマップシステムの核となる。まず視線方向によって，中心とその周りのどのブロックを表示するかを決定するのがREL_WAY_DATAの役割。視線にかかわらず，表示される可能性のあるブロックは視点を中心とした9×9ブロックとなっている。この例ではブロックの1辺が2048であるために，そこそこ遠くまで見えるようなサイズとして考えている。

そして，この9×9ブロックから視線方向に16段階で表示するブロックを選び出す。それぞれのブロックのオフセット位置は，計算で求めることができる。だが効率を追求するならば，オフセット位置はデータとしてもっておこう。それがREL_MAP_DATAとなっている。これによってオフセット位置を計算しながらブロックを表示していくことになる。

処理の流れは，以下のようになるだろう。

1) 視点座標により，中心ブロックを決定
2) 視線方向によって表示ブロックを決定
3) 表示ブロックのオフセットを決定，表示

本来ならば表示のときにディテールレベル，さらに空気遠近法を取り入れるのだが，このリストではそこまではやっていない。ディテールを変えることが効率化のポイント，とかいっておきながらやっていないのは少々情けない。というのもテストとなるデータがなかったのである（掲載するスペースの都

図3　マップデータの構成

マップデータはメモリに配列の形としてもつ。サンプルでは，

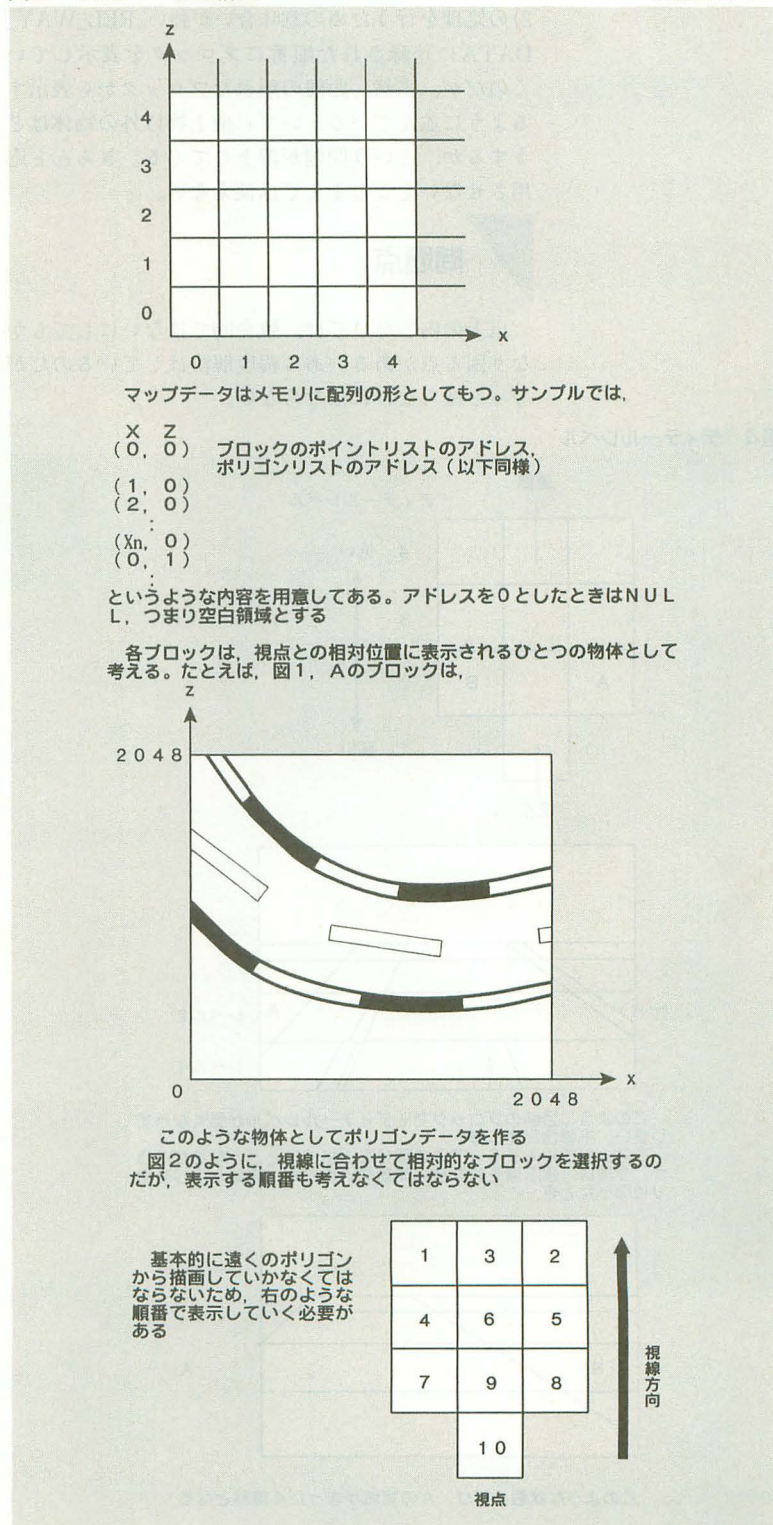
X　Z
(0，0)　ブロックのポイントリストのアドレス，ポリゴンリストのアドレス（以下同様）
(1，0)
(2，0)
:
(Xn，0)
(0，1)
:

というような内容を用意してある。アドレスを0としたときはNULL，つまり空白領域とする

各ブロックは，視点との相対位置に表示されるひとつの物体として考える。たとえば，図1，Aのブロックは，

このような物体としてポリゴンデータを作る

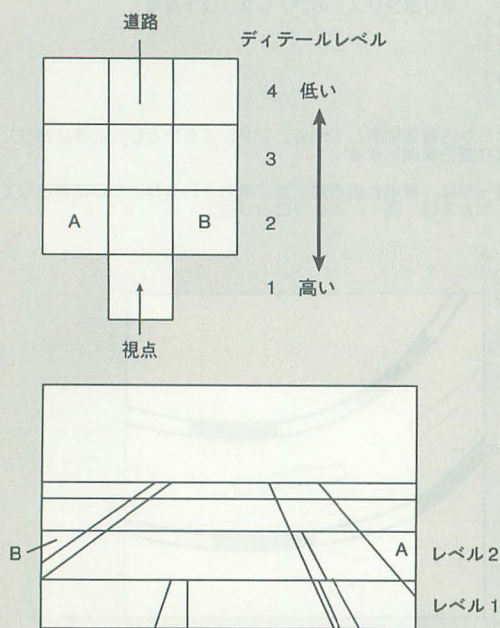
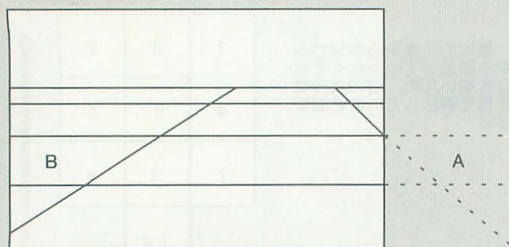
図2のように，視線に合わせて相対的なブロックを選択するのだが，表示する順番も考えなくてはならない

基本的に遠くのポリゴンから描画していかなくてはならないため，右のような順番で表示していく必要がある

合もあるが)。まあ，テストデータぐらいは簡単にできるであろうから，ディテールレベルのデータだけ内部に用意してある。

さて，ここで注意するのは各ブロックの優先順位。データとしてREL_WAY_DATAをもっていたのは，2)の処理を行うための意味合いが強い。REL_WAY_DATAに登録された順番にブロックを表示していくのだが，当然，距離の離れたブロックから表示するように並んでいる。いずれ地上物以外の物体はどうするか，という問題が浮上してくる。きちんと応用させないとこのままでは使えない。

問題点

以上の内容だけでは，致命的ではないにしてもかなり困る点がある。ある程度解決はしているのだがもっと練らなければならない。

図4　ディテールレベル

このように目前のブロックでもディテールレベルが低くなってしまい，不都合が生じる
では，レベルをずらして2の部分を1にするとどうなるだろうか。今度は，表示はそれなりに問題ないのだが，視線が少しB寄りになったとき，

このような状態となり，Aの領域がまったく無駄となる

問題のひとつは，どのブロックまでを最高のディテールで表示するかである(図4)。ディテールを落とすときには，しきい値によってどのディテールで表示するか決定する。この方法だとしきい値を境にほんのわずかな距離でもディテールを落とすことになってしまう。これはディテールを落とす距離をブロック1つ分遠くにすればよいと思うが，下手をすると図4のように無駄をさらに増大させる可能性を秘めている。

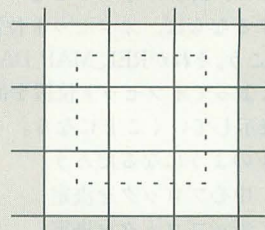
また，ブロックの頂点付近に視点があるときはどうであろう。当然複数のブロックが視点付近に接近している。当然これらのブロックは精密な描写をする必要があり，ディテールレベルは最高。となると，これらのブロックの大部分を占める表示されない物体のために，先ほどと同じように無駄な計算をしなくてはならない。

結局，頂点付近に視点があるときはブロックのほとんどが無駄となり，逆にブロックの中心付近に存在するときは周りのブロックが無駄になってしまうのが痛い。ということで解決方法も考える。それは，ブロックの端を重複させればかなりのごまかしが可能となるのだ。図5のような区切り方をすれば無駄なブロックまで表示にもち込むことはなくなる。

しかし，この方法もバランスが微妙で，下手にやると1つひとつのブロックが膨大になってしまう可能性がある。

そして，ブロック自体の大きさがどのように影響するだろうか。当然ひとつのブロックが大きければ，メモリサイズは同じでも地形全体はより広くなる。サンプルリストではひとつのブロックのサイズを2048×2048，そしてマップ全体を64×64ブロックとしているが，それだけでもかなりのメモリを食う。

まず，各ブロックが独立したオブジェクトになっているとすると，ブロックのポインタだけで64×64×4バイト。ひとつのブロックで座標リストとポリゴンリスト，それと各ディテールレベルが必要。これが64×64存在するだけで，マップデータは2Mバイトは軽く超えることになる。そこそこのサイズを実

図5　ブロックの領域

実際のブロックは実線で区切られる。しかし，実際のデータでは，点線領域まで含まれる。つまり，必ずしもデータをブロックのサイズに収める必要はないのである

現しようとしても，すぐに許されないサイズになってしまうのである。実際はすべてのブロックが独立して埋まっているような作りはしないので，ここまで極端ではないのだが。

大きすぎるのがまずいなら，と安易にブロックのサイズを縮めると，当然のことながらマップ全体の大きさが小さくなってしまう。結局のところ，地形全体をサポートするためには膨大なメモリを食うのだ。さらに，視野を確保するためには多くのブロックを表示しなくてはならず，さらにまた問題が生じてくる。たとえば道路などをブロックで区切ると，そのブロックの境で無駄な頂点を増やすことになってしまうのだ。この方法に，先ほどのブロックを重複して区切る方法を加えると表示面積の増加もあり，問題がぼろぼろ出てくることがわかる。

逆に1ブロックのサイズを大きくすると今度は無駄な計算を増やすことになる。すでにわかっているが，ブロックの頂点付近でそのブロックの中心と反対方向を見ているときにはそのブロック（ディテールレベルは最高であろう）はほとんど無駄となってしまう。これは1ブロックが小さければそれほど問題にはならないのだ。

ということでどちらに傾けても問題が生じるのである。これを解決するのはバランスのみ。作りたいモノによって変化させなければならないのは，はっきりしたであろう。

当然，このようなマップ構成以外も考えられる。そのときには長方形の領域で収まらないデータも考えなければならない。高速道路なんかはその最大の例ではないだろうか。長方形にエリアを取るとほとんどが空白領域になってしまうことになるので，また別のデータ構造にしなくてはならない。

終わりに

ひととおり，マップシステムについて考えてきたが，結論としてやはりマップエディタが必要になってきた。しばらくはマップエディタを作ることになるだろう。それにともなって，いよいよC言語を学ぶときがきたのかもしれない。そうだな，アセンブラ1本でもいいのだが，そろそろ衣更えということで真剣に検討してみるか。それではまた。

リスト

```
  1:
  2: mp_size_x       equ     2048    *マップピクセルXサイズ
  3: mp_size_z       equ     2048    *マップピクセルZサイズ
  4: sqrl            equ     64      *1辺ブロック数
  5: PR_MAP:
  6:         lea.l   ROTDATA(pc),a6  *3Dパラメータ
  7:         lea.l   MATRIX(pc),a2   *行列リスト
  8:         move.w  6(a6),d5        *オイラー角X
  9:         move.w  8(a6),d6        *オイラー角Y
 10:         move.w  10(a6),d7       *オイラー角Z
 11:         jsr     T_GETMATRIXview *変換行列を"MATRIX"に保存
 12:         lea.l   ROTDATA(pc),a6
 13:         move.w  6(a6),d5        *オイラー角X
 14:         move.w  8(a6),d6        *オイラー角Y
 15:         move.w  10(a6),d7       *オイラー角Z
 16:         move.w  12(a6),d0       *光源α
 17:         move.w  14(a6),d1       *光源β
 18:         jsr     T_GETRAY
 19:         lea.l   MATRIX(pc),a2   *"MATRIX"に光源データ登録
 20:         move.w  d1,18(a2)
 21:         move.w  d0,20(a2)
 22:         move.l  d2,22(a2)
 23:         lea.l   ROTDATA(pc),a6
 24:         lea.l   MAP_WORK(pc),a5  *一応マップ用のワークを
 25:         lea.l   DUM_POINT(pc),a4 *頂点数0のポイントリストを変換
 26:         lea.l   MATRIX(pc),a2
 27:         jsr     T_TRANSLATER_MATRIXview *行列をSLASH内にセットする
 28:         move.l  PX(pc),d0                *視点座標X
 29:         move.l  PZ(pc),d1                *視点座標Z
 30:         move.l  PY(pc),d6                *視点座標Y
 31:         move.w  d6,HY                    *Y座標はそのまま
 32:         move.w  d0,d4
 33:         move.w  d1,d5
 34:         and.w   #mp_size_x-1,d4
 35:         and.w   #mp_size_z-1,d5
 36:         neg.w   d4              *d4=my x(0--2047)微細なオフセット値
 37:         neg.w   d5              *d5=my z(0--2047)
 38:         moveq.l #11,d2
 39:         lsr.l   d2,d0           */2048  マップ座標X
 40:         lsr.l   d2,d1           */2048  マップ座標Zにする
 41:         lea.l   MAPAREA,a0      *実際のマップデータ
 42:                         *周り9*9ブロックと中心との相対座標リスト
 43:         lea.l   REL_MAP_DATA(pc),a1
 44:         move.w  VY(pc),d2 *オイラー角Y（方向を示す）
 45:         addi.w  #$80,d2   *4捨5入のための補正
 46:         move.w  VX(pc),d3 *ピッチによってひっくり返ったときは方向反転
 47:         sub.w   #1024,d3
 48:         cmp.w   #2048,d3
 49:         bcs     maprev
 50:         sub.w   #$100*8,d2
 51: maprev:
 52:         add.w   #$100*8,d2
 53:         andi.w  #$f00,d2
 54:         ror.w   #6,d2                   *0--15
 55:                         *4      方向によって16段階
 56:                         *方向により、どのブロックが表示されるかを決定
 57:         lea.l   REL_WAY_DATA(pc),a2
 58:         adda.w  d2,a2
 59:         movea.l (a2),a2
 60:         move.w  #sqrl-1,d3 *ANDでマップ座標内にするための値
 61: pr_maploop:
 62:         move.w  (a2)+,d2   *周り9*9ブロックのうちの番号
 63:         bmi     prmapend   *マイナスで終了
 64:         move.w  (a2)+,d6   *ディテールレベル
 65:         movem.l d0-d1/d3-d5/a0-a2,-(sp)          *レジスタプッシュ
 66:         lsl.w   #3,d2      *16bite      1rel_map
 67:         adda.w  d2,a1      *そのブロックと中心との相対距離をポイント
 68:         add.w   (a1)+,d0   *マップ座標X
 69:         add.w   (a1)+,d1   *マップ座標Z
 70:         and.w   d3,d0      *端を越えたら反対を
 71:         and.w   d3,d1
 72:         lsl.w   #8,d1      **256 map    xsize=64*1LongWord
 73:         lsl.w   #2,d0      **4
 74:         add.w   d0,d1
 75:         move.l  0(a0,d1.w),d0   *そのブロックの実データ
 76:         beq     maper_next      *ピクセル空なら飛ばす
 77:         add.w   (a1)+,d4        *ブロックを表示する時の相対座標
 78:         add.w   (a1)+,d5
 79:         move.w  d4,HX           *変換オフセット座標としてセット
 80:         move.w  d5,HZ
 81:         movea.l d0,a0           *ブロックデータ
 82:         lea.l   ROTDATA(pc),a6
 83:         lea.l   MAP_WORK(pc),a5 *マップ用変換ワーク
 84:         movea.l (a0)+,a4        *ポイントリスト
 85:         move.l  a0,-(sp)
 86:         jsr     T_TRANSLATEREPview      *座標変換
 87:         move.l  (sp)+,a0
 88:         movea.l (a0)+,a6                *ポリゴンリスト
 89:         lea.l   MAP_WORK(pc),a5
 90:         jsr     T_DRAWPOLY              *1ブロック表示
 91: maper_next:
 92:         movem.l (sp)+,d0-d1/d3-d5/a0-a2 *レジスタポップ
 93:         bra     pr_maploop              *次のブロックへ
 94: prmapend:
 95:                 rts
 96:
 97: *周り9*9ブロックに0～80の番号を与える
 98:
 99: mpd00   equ     0
100: mpd01   equ     1
101: mpd02   equ     2
102: mpd03   equ     3
103: mpd04   equ     4
104: mpd05   equ     5
105: mpd06   equ     6
106: mpd07   equ     7
107: mpd08   equ     8
108: mpd10   equ     0+9
109: mpd11   equ     1+9
110: mpd12   equ     2+9
111: mpd13   equ     3+9
112: mpd14   equ     4+9
113: mpd15   equ     5+9
114: mpd16   equ     6+9
115: mpd17   equ     7+9
116: mpd18   equ     8+9
117: mpd20   equ     0+9*2
118: mpd21   equ     1+9*2
119: mpd22   equ     2+9*2
120: mpd23   equ     3+9*2
121: mpd24   equ     4+9*2
122: mpd25   equ     5+9*2
123: mpd26   equ     6+9*2
124: mpd27   equ     7+9*2
125: mpd28   equ     8+9*2
126: mpd30   equ     0+9*3
127: mpd31   equ     1+9*3
128: mpd32   equ     2+9*3
129: mpd33   equ     3+9*3
130: mpd34   equ     4+9*3
131: mpd35   equ     5+9*3
132: mpd36   equ     6+9*3
```

```
 133: mpd37    equ      7+9*3
 134: mpd38    equ      8+9*3
 135: mpd40    equ      0+9*4
 136: mpd41    equ      1+9*4
 137: mpd42    equ      2+9*4
 138: mpd43    equ      3+9*4
 139: mpd44    equ      4+9*4
 140: mpd45    equ      5+9*4
 141: mpd46    equ      6+9*4
 142: mpd47    equ      7+9*4
 143: mpd48    equ      8+9*4
 144: mpd50    equ      0+9*5
 145: mpd51    equ      1+9*5
 146: mpd52    equ      2+9*5
 147: mpd53    equ      3+9*5
 148: mpd54    equ      4+9*5
 149: mpd55    equ      5+9*5
 150: mpd56    equ      6+9*5
 151: mpd57    equ      7+9*5
 152: mpd58    equ      8+9*5
 153: mpd60    equ      0+9*6
 154: mpd61    equ      1+9*6
 155: mpd62    equ      2+9*6
 156: mpd63    equ      3+9*6
 157: mpd64    equ      4+9*6
 158: mpd65    equ      5+9*6
 159: mpd66    equ      6+9*6
 160: mpd67    equ      7+9*6
 161: mpd68    equ      8+9*6
 162: mpd70    equ      0+9*7
 163: mpd71    equ      1+9*7
 164: mpd72    equ      2+9*7
 165: mpd73    equ      3+9*7
 166: mpd74    equ      4+9*7
 167: mpd75    equ      5+9*7
 168: mpd76    equ      6+9*7
 169: mpd77    equ      7+9*7
 170: mpd78    equ      8+9*7
 171: mpd80    equ      0+9*8
 172: mpd81    equ      1+9*8
 173: mpd82    equ      2+9*8
 174: mpd83    equ      3+9*8
 175: mpd84    equ      4+9*8
 176: mpd85    equ      5+9*8
 177: mpd86    equ      6+9*8
 178: mpd87    equ      7+9*8
 179: mpd88    equ      8+9*8
 180:
 181: *視線の方向によって表示すべきブロックを指定
 182: *(周り9*9ブロックのうちどれか)
 183: REL_WAY_DATA:
 184:          dc.l  RWD00,RWD01,RWD02,RWD03,RWD04,RWD05,RWD06,RWD07
 185:          dc.l  RWD08,RWD09,RWD0a,RWD0b,RWD0c,RWD0d,RWD0e,RWD0f
 186: RWD00:
 187:          dc.w  mpd04,0,mpd13,0,mpd15,0,mpd14,0,mpd23,0,mpd25,0,
mpd24,0
 188:          dc.w  mpd33,0,mpd35,0,mpd34,0,mpd43,0,mpd45,0,mpd44,0
 189:          dc.w  -1,-1
 190: RWD01:
 191:          dc.w  mpd02,0,mpd03,0,mpd12,0,mpd13,0,mpd14,0,mpd22,0,
mpd23,0,mpd24,0
 192:          dc.w  mpd33,0,mpd34,0,mpd35,0,mpd43,0,mpd45,0,mpd44,0
 193:          dc.w  -1,-1
 194: RWD02:
 195:          dc.w  mpd11,0,mpd12,0,mpd21,0,mpd13,0,mpd31,0,mpd22,0,
mpd23,0,mpd32,0
 196:          dc.w  mpd24,0,mpd42,0,mpd33,0,mpd34,0,mpd43,0,mpd44,0
 197:          dc.w  -1,-1
 198: RWD03:
 199:          dc.w  mpd20,0,mpd30,0,mpd21,0,mpd31,0,mpd41,0,mpd22,0,
mpd32,0,mpd42,0
 200:          dc.w  mpd33,0,mpd43,0,mpd53,0,mpd34,0,mpd54,0,mpd44,0
 201:          dc.w  -1,-1
 202: RWD04:
 203:          dc.w  mpd40,0,mpd31,0,mpd51,0,mpd41,0,mpd32,0,mpd52,0,
mpd42,0,mpd33,0
 204:          dc.w  mpd53,0,mpd43,0,mpd34,0,mpd54,0,mpd44,0
 205:          dc.w  -1,-1
 206: RWD05:
 207:          dc.w  mpd60,0,mpd50,0,mpd61,0,mpd51,0,mpd41,0,mpd62,0,
mpd52,0,mpd42,0
 208:          dc.w  mpd53,0,mpd43,0,mpd33,0,mpd54,0,mpd34,0,mpd44,0
 209:          dc.w  -1,-1
 210: RWD06:
 211:          dc.w  mpd71,0,mpd72,0,mpd61,0,mpd73,0,mpd51,0,mpd62,0,
mpd63,0,mpd52,0
 212:          dc.w  mpd64,0,mpd42,0,mpd53,0,mpd54,0,mpd43,0,mpd44,0
 213:          dc.w  -1,-1
 214: RWD07:
 215:          dc.w  mpd82,0,mpd83,0,mpd72,0,mpd73,0,mpd74,0,mpd62,0,
mpd63,0,mpd64,0
 216:          dc.w  mpd53,0,mpd54,0,mpd55,0,mpd43,0,mpd45,0,mpd44,0
 217:          dc.w  -1,-1
 218: RWD08:
 219:          dc.w  mpd84,0,mpd73,0,mpd75,0,mpd74,0,mpd63,0,mpd65,0,
mpd64,0,mpd53,0
 220:          dc.w  mpd55,0,mpd54,0,mpd43,0,mpd45,0,mpd44,0
 221:          dc.w  -1,-1
 222: RWD09:
 223:          dc.w  mpd86,0,mpd85,0,mpd76,0,mpd75,0,mpd74,0,mpd66,0,
mpd65,0,mpd64,0
 224:          dc.w  mpd55,0,mpd54,0,mpd53,0,mpd45,0,mpd43,0,mpd44,0
 225:          dc.w  -1,-1
 226: RWD0a:
 227:          dc.w  mpd77,0,mpd76,0,mpd67,0,mpd75,0,mpd57,0,mpd66,0,
mpd65,0,mpd56,0
 228:          dc.w  mpd64,0,mpd46,0,mpd55,0,mpd54,0,mpd45,0,mpd44,0
 229:          dc.w  -1,-1
 230: RWD0b:
 231:          dc.w  mpd68,0,mpd58,0,mpd67,0,mpd57,0,mpd47,0,mpd66,0,
mpd56,0,mpd46,0
 232:          dc.w  mpd55,0,mpd45,0,mpd35,0,mpd54,0,mpd34,0,mpd44,0
 233:          dc.w  -1,-1
 234: RWD0c:
 235:          dc.w  mpd48,0,mpd37,0,mpd57,0,mpd47,0,mpd36,0,mpd56,0,
mpd46,0,mpd35,0
 236:          dc.w  mpd55,0,mpd45,0,mpd34,0,mpd54,0,mpd44,0
 237:          dc.w  -1,-1
 238: RWD0d:
 239:          dc.w  mpd28,0,mpd38,0,mpd27,0,mpd37,0,mpd47,0,mpd26,0,
mpd36,0,mpd46,0
 240:          dc.w  mpd35,0,mpd45,0,mpd55,0,mpd34,0,mpd54,0,mpd44,0
 241:          dc.w  -1,-1
 242: RWD0e:
 243:          dc.w  mpd17,0,mpd16,0,mpd27,0,mpd15,0,mpd37,0,mpd26,0,
mpd25,0,mpd36,0
 244:          dc.w  mpd24,0,mpd46,0,mpd35,0,mpd34,0,mpd45,0,mpd44,0
 245:          dc.w  -1,-1
 246: RWD0f:
 247:          dc.w  mpd06,0,mpd05,0,mpd16,0,mpd15,0,mpd14,0,mpd26,0,
mpd25,0,mpd24,0
 248:          dc.w  mpd35,0,mpd34,0,mpd33,0,mpd45,0,mpd43,0,mpd44,0
 249:          dc.w  -1,-1
 250:
 251: *周り9*9(計81)ブロックそれぞれと
 252: *中心との相対座標のデータ
 253:
 254: REL_MAP_DATA:
 255:          dc.w   -4                    *マップオフセットX
 256:          dc.w   4                     *マップオフセットZ
 257:          dc.w   -mp_size_x*4          *オフセット距離X
 258:          dc.w   mp_size_z*4           *オフセット距離Z
 259:          dc.w   -3,4,-mp_size_x*3,mp_size_z*4
 260:          dc.w   -2,4,-mp_size_x*2,mp_size_z*4
 261:          dc.w   -1,4,-mp_size_x*1,mp_size_z*4
 262:          dc.w   0,4,mp_size_x*0,mp_size_z*4
 263:          dc.w   1,4,mp_size_x*1,mp_size_z*4
 264:          dc.w   2,4,mp_size_x*2,mp_size_z*4
 265:          dc.w   3,4,mp_size_x*3,mp_size_z*4
 266:          dc.w   4,4,mp_size_x*4,mp_size_z*4
 267:          dc.w   -4,3,-mp_size_x*4,mp_size_z*3
 268:          dc.w   -3,3,-mp_size_x*3,mp_size_z*3
 269:          dc.w   -2,3,-mp_size_x*2,mp_size_z*3
 270:          dc.w   -1,3,-mp_size_x*1,mp_size_z*3
 271:          dc.w   0,3,-mp_size_x*0,mp_size_z*3
 272:          dc.w   1,3,mp_size_x*1,mp_size_z*3
 273:          dc.w   2,3,mp_size_x*2,mp_size_z*3
 274:          dc.w   3,3,mp_size_x*3,mp_size_z*3
 275:          dc.w   4,3,mp_size_x*4,mp_size_z*3
 276:          dc.w   -4,2,-mp_size_x*4,mp_size_z*2
 277:          dc.w   -3,2,-mp_size_x*3,mp_size_z*2
 278:          dc.w   -2,2,-mp_size_x*2,mp_size_z*2
 279:          dc.w   -1,2,-mp_size_x*1,mp_size_z*2
 280:          dc.w   0,2,mp_size_x*0,mp_size_z*2
 281:          dc.w   1,2,mp_size_x*1,mp_size_z*2
 282:          dc.w   2,2,mp_size_x*2,mp_size_z*2
 283:          dc.w   3,2,mp_size_x*3,mp_size_z*2
 284:          dc.w   4,2,mp_size_x*4,mp_size_z*2
 285:          dc.w   -4,1,-mp_size_x*4,mp_size_z*1
 286:          dc.w   -3,1,-mp_size_x*3,mp_size_z*1
 287:          dc.w   -2,1,-mp_size_x*2,mp_size_z*1
 288:          dc.w   -1,1,-mp_size_x*1,mp_size_z*1
 289:          dc.w   0,1,mp_size_x*0,mp_size_z*1
 290:          dc.w   1,1,mp_size_x*1,mp_size_z*1
 291:          dc.w   2,1,mp_size_x*2,mp_size_z*1
 292:          dc.w   3,1,mp_size_x*3,mp_size_z*1
 293:          dc.w   4,1,mp_size_x*4,mp_size_z*1
 294:          dc.w   -4,0,-mp_size_x*4,mp_size_z*0
 295:          dc.w   -3,0,-mp_size_x*3,mp_size_z*0
 296:          dc.w   -2,0,-mp_size_x*2,mp_size_z*0
 297:          dc.w   -1,0,-mp_size_x*1,mp_size_z*0
 298:          dc.w   0,0,mp_size_x*0,mp_size_z*0
 299:          dc.w   1,0,mp_size_x*1,mp_size_z*0
 300:          dc.w   2,0,mp_size_x*2,mp_size_z*0
 301:          dc.w   3,0,mp_size_x*3,mp_size_z*0
 302:          dc.w   4,0,mp_size_x*4,mp_size_z*0
 303:          dc.w   -4,-1,-mp_size_x*4,-mp_size_z*1
 304:          dc.w   -3,-1,-mp_size_x*3,-mp_size_z*1
 305:          dc.w   -2,-1,-mp_size_x*2,-mp_size_z*1
 306:          dc.w   -1,-1,-mp_size_x*1,-mp_size_z*1
 307:          dc.w   0,-1,-mp_size_x*0,-mp_size_z*1
 308:          dc.w   1,-1,mp_size_x*1,-mp_size_z*1
 309:          dc.w   2,-1,mp_size_x*2,-mp_size_z*1
 310:          dc.w   3,-1,mp_size_x*3,-mp_size_z*1
 311:          dc.w   4,-1,mp_size_x*4,-mp_size_z*1
 312:          dc.w   -4,-2,-mp_size_x*4,-mp_size_z*2
 313:          dc.w   -3,-2,-mp_size_x*3,-mp_size_z*2
 314:          dc.w   -2,-2,-mp_size_x*2,-mp_size_z*2
 315:          dc.w   -1,-2,-mp_size_x*1,-mp_size_z*2
 316:          dc.w   0,-2,mp_size_x*0,-mp_size_z*2
 317:          dc.w   1,-2,mp_size_x*1,-mp_size_z*2
 318:          dc.w   2,-2,mp_size_x*2,-mp_size_z*2
 319:          dc.w   3,-2,mp_size_x*3,-mp_size_z*2
 320:          dc.w   4,-2,mp_size_x*4,-mp_size_z*2
 321:          dc.w   -4,-3,-mp_size_x*4,-mp_size_z*3
 322:          dc.w   -3,-3,-mp_size_x*3,-mp_size_z*3
 323:          dc.w   -2,-3,-mp_size_x*2,-mp_size_z*3
 324:          dc.w   -1,-3,-mp_size_x*1,-mp_size_z*3
 325:          dc.w   0,-3,-mp_size_x*0,-mp_size_z*3
 326:          dc.w   1,-3,mp_size_x*1,-mp_size_z*3
 327:          dc.w   2,-3,mp_size_x*2,-mp_size_z*3
 328:          dc.w   3,-3,mp_size_x*3,-mp_size_z*3
 329:          dc.w   4,-3,mp_size_x*4,-mp_size_z*3
 330:          dc.w   -4,-4,-mp_size_x*4,-mp_size_z*4
 331:          dc.w   -3,-4,-mp_size_x*3,-mp_size_z*4
 332:          dc.w   -2,-4,-mp_size_x*2,-mp_size_z*4
 333:          dc.w   -1,-4,-mp_size_x*1,-mp_size_z*4
 334:          dc.w   0,-4,mp_size_x*0,-mp_size_z*4
 335:          dc.w   1,-4,mp_size_x*1,-mp_size_z*4
 336:          dc.w   2,-4,mp_size_x*2,-mp_size_z*4
 337:          dc.w   3,-4,mp_size_x*3,-mp_size_z*4
 338:          dc.w   4,-4,mp_size_x*4,-mp_size_z*4
 339: ROTDATA:                        *変換3Dパラメータ
 340: HX:      dc.w   0
 341: HY:      dc.w   0
 342: HZ:      dc.w   0
 343: VX:      dc.w   0
 344: VY:      dc.w   0
 345: VZ:      dc.w   0
 346:          dc.w   0
 347:          dc.w   32
 348:          dc.w   0
 349:          dc.w   0
 350: DUM_POINT:                      *頂点数0のポイントリスト
 351:          dc.w   0
 352: DUM_MINMAX:
 353:          ds.w   4*16+1
 354: MATRIX:
 355:          ds.w   32
```

THE SENTINEL

〈対応機種一覧〉●MZ-80 K/C/700/1500●MZ-80 B/2000●MZ-2500/2861●X1●X1 turbo/Z●PC-8001/8801/88●SMC-777/C●PASOPIA/5●PASOPIA/7●FM-7/77/AV●MSX/2/2+/turbo R●PC-286/386/486/9801/98/9821●X68000/X68030
掲載されたプログラムの利用には各機種用のS-OS "SWORD" システムが必要です。

第145部　YGCS ver.0.30

●やっとダンプリストの掲載だ!

まず，最初に断わっておきますが，今月掲載した「YGCS ver.0.30」を使いこなすためには，1994年2月号に掲載された「YGCS ver.0.20リファレンス」が必要となります。

今月掲載したのは「YGCS ver.0.30」のダンプリスト，そして，サンプルリストのソースとダンプリストのみです。なお，ダンプリスト掲載までの間，ちょっとだけバージョンアップしたので，その拡張コールのリファレンスも掲載してあります。

しかし，サンプルリストだけでは，このシステムを使いこなすことはできません。当たり前ですね。とりあえずサンプルを見れば，ある程度，データ構造やコールの使い方の予測はつきます。が，すべてのコールの使用例を示しているわけでもないし，注釈だけでは，どんなデータをどのように用意すればいいのか見当もつきません。

まあ，ダンプリストの掲載を見合わせたのは，この「YGCS ver.0.30」がまだ完成したものではないため，一度読者の皆さんにモニタになってもらい，さらに完成度を上げたものを掲載しよう，という意図がありました。しかし，システムを使いこなすまでの時間が足りなかったためか，それともこのままのシステムでなんの問題もなかったのかわかりませんが，あまり皆さんの意見を反映できなかったのは残念です。

ということで，この「YGCS ver.0.30」を使うためには2月号が必要となります。しかし，2月号をもっていない人には使えないよ，なんてことはいえないので，ここで，「YGCS ver.0.20リファレンス」のコピーサービスを行います。

アンケートハガキの「編集室へのメッセージ」の欄に，

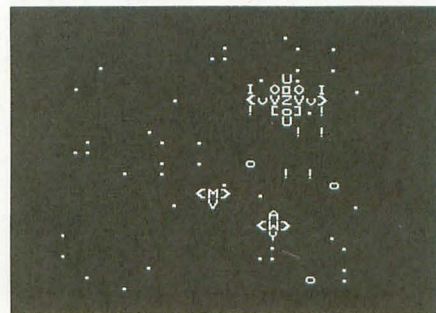
「YGCS ver.0.20リファレンス」
コピーサービス希望

と明記のうえ，蛍光マーカーかなにかで目立つようにしてください。できるかぎり迅速に対応したいところですが，応募人数によっては多少遅くなります。

●10年といったらひと昔

本当に10年たちました。え，もちろん全機種共通システムが始まってからですよ。いまとなっては，あまり元気のなくなってしまったTHE SENTINELですが，まだ死ねません。

で，THE SENTINELの今後の予定としては，「YGCS ver.0.30」をベースにした「シューティングゲーム作成講座」を行います。一応，「YGCS ver.0.30」をベースに，シューティングゲームを作っていこうという内容になりそうです。

ということで，S-OS "SWORD" は使っていないという人でもシューティングゲームを作っていてわからないことがあれば，質問をお寄せください。S-OS上の記事ということで，X68000のスプライトの扱い方とか，ラスターがどうとか，というような質問には答えられませんが，うまいデータ構造の作り方などでちょっと悩んでいるところがあれば，質問内容を明記してアンケートハガキか官製ハガキでお送りください。

それから，以前，なんとか掲載したいといっていた筑紫氏の「Z80テクニック」も，「シューティングゲーム作成講座」と一緒に掲載していくかもしれません。忘れているわけではありせんので，ご安心を。

そうそう，最近ちょっとしたシミュレータプログラムが投稿されてきました。それなりに面白い試みですが，実際に利用価値があるかどうかは微妙なところ。現在チェック中です。もしかしたら掲載できるかもしれませんので，埼玉県の平野さん，楽しみにしていてください。

最後になりましたが，今月掲載した「YGCS ver.0.30」は掲載した時点でコピーフリーとします。ということで，4月号から始まった「アプリケーションのフリーソフト化計画」のリストに，

1994年6月号
第145部　YGCS ver.0.30

が加わります。相沢さん，ご協力感謝します。

なお，「アプリケーションのフリーソフト化計画」では，引き続き作者からの連絡をお待ちしています。もしも，この記事が目に止まったら，ぜひともアンケートハガキでご連絡をお願いします。

1994■インデックス

■94年1月号
第139部　S-OSで学ぶZ80マシン語講座(2)
■94年2月号
第140部　YGCSver.0.20ユーザーズマニュアル
第141部　S-OSで学ぶZ80マシン語講座(3)
■94年3月号
第142部　S-OSで学ぶZ80マシン語講座(4)
■94年4月号
第143部　S-OSで学ぶZ80マシン語講座(5)
■94年5月号
第144部　S-OSで学ぶZ80マシン語講座(6)

全機種共通
S-OS"SWORD"要

Yu-ri Game Core System
YGCS ver.0.30

Aizawa Eiki
相沢 栄樹

今月は「YGCS ver.0.30」のダンプリストとサンプルプログラムを掲載します。来月から，このシステムを使った「シューティングゲーム作成講座」が始まる予定です。がんばって打ち込みましょう。

今回は，「YGCS ver.0.30」のダンプリストと実際にシステムを使用したサンプルプログラムの説明をします。リストとマップ，さらに1994年2月号に掲載したユーザーズマニュアルを参照しながら読んでみてください。

なお，リファレンスを掲載したときより，ちょっとだけバージョンアップしました。それにともない追加コールもありますが，たいしたものではないので簡単に理解できるでしょう。そして，本システムおよびサンプル作成には，アセンブラとしてWZDを使用しています。ほかのアセンブラを使っている人は，適宜対応してください。

(1)初期設定

まず始めにすべきことはプログラムの初期化です。これをしておかないと，あとあと困ります(当たり前ですね)。[COLD_]から[HOT_]までのところがシステム関係の初期化の部分で，そのあとユーザープログラム等の初期化を行っています。

システムの初期化の部分は，

1) IDテーブルの登録
2) キャラ出現スケジュールの登録
3) キーテーブルの登録

の順番になっています。これらの処理を行わないと，システムが正常に機能しなくなるので注意してください。

ユーザープログラムの初期化では，スコアの初期化のみしています。

(2)メイン制御

「YGCS ver.0.30」において，メイン制御というのはなきに等しい部分です。サンプルプログラムでも，ほんの数行に収まっています。詳しくはユーザーズマニュアルを参照するといいでしょう(お，手抜き)。

(3)自機処理

とりあえず自機処理の見本(お手本ではない)です。本当に基本的な部分のみですが，自分でプログラムする際の参考にはなると思います。

まず，動作モード分岐処理の部分は見てのとおり2つしかありません。モードが増えた場合はテーブルに追加してください。

次に初期化の部分です。ワークについての詳しい説明はユーザーズマニュアルを参照してもらうとして，実際，すべてのワークを初期化する必要はありません。このプログラムでも，必要最小限の初期化にとどめています。初期化部分の最後のところで，MODを1にしてリターンしています。この処理によって次回からはモード1のプログラムに処理が分岐します。

モード1の部分は，自機のメイン制御部です。やっていることといえば，

1) 当たりの判定(敵，敵弾との)
2) 各種処理のコール
   - 自機の移動
   - 移動範囲の制限
   - 弾の発射

これぐらいで，この部分はあまり重要ではありません。ちなみに最後で行っている処理は，自機の座標をグローバル変数に書き込んでいるだけで，ほかのキャラが自機の位置を確認する際に必要になります。

さて，次の移動処理ではちょっとしたテクニックを使用しています。読み込んできた方向キーのデータを，移動量テーブルへのポインタとして使用していることです。この方法では条件判断をしなくていいので，美しいプログラムが書けるという利点もあります(本当か?)。データを2倍して利用しているのは，XYの2バイトデータへのオフセットだからです。

移動範囲の制限については，これといって解説することはありません。

弾発射処理の冒頭で行っているのは，ワークの空きが2個以上あるのを確認する処理です。これにより，左右のどちらかの弾が出続けるという，間抜けな事態を回避しています。そのあとで，左右2つの弾を定義(発生)しています。定義方法を確認しておいてください。

さて最後は当たり処理です。このサンプルでは自機が光るだけなので，自機の絵のパターンを入れ替えて，ヒットポイントを元に戻しているだけです。

入れ替えた絵は，次回モード1の処理で元に戻されます。

(4)ボス敵処理

大まかな部分は自機の処理と同じです。

この処理で特に注目してほしいのは，動きによりモードを切り替えているところと，サイン関数による円弧運動の部分です。ユーザーズマニュアルと合わせて見てもらえれば，割と簡単に理解してもらえると思います。

自機の処理と大幅に違うのが，当たり処理です。このボスの場合は，即死亡(爆発)になります。よって，爆発を定義(発生)

したあとに点数を入れ，自分そのものを消去しているわけです。

(5)敵弾処理

この弾は自機に向かって発射される弾です。よって，初期設定のところで自機の方向を割り出し，移動量を計算しています。ここで算出された移動量をもとに，弾は移動していくことになります。そして，移動したあとに画面外判定をして，範囲外であれば消去します。簡単ですね。

以上で大まかな説明を終わりますが，ここで説明しなかった部分に関しても，マニュアルを見ながら目を通しておいてください。ひととおりの使い方がわかると思います。

では，がんばって使ってやってください。

表1 サンプルリストの内容

[全体マップ]	
15 - 16 : ヘッダ読み込み	
21 - 35 : 初期設定	(1)
36 - 48 : メイン制御	(2)
53 - 67 : スコアの初期化	
72 - 98 : 背景の全体描画	
103 - 112 : スコア表示	
117 - 139 : 背景スクロール＆描画	
144 - 311 : 自機処理	(3)
316 - 374 : 自機弾処理	
379 - 448 : ザコ敵処理	
453 - 588 : ボス敵処理	(4)
593 - 646 : 敵弾処理	(5)
651 - 716 : 爆発処理	
721 - 734 : ザコ爆発セット	
739 - 792 : ボス爆発セット	
797 - 807 : IDテーブル	
812 - 830 : スケジュールテーブル	
835 - 837 : キーテーブル	
842 - 845 : 全体変数	

[自機マップ]
144 - 149 : 動作モード分岐処理
153 - 168 : 初期化
172 - 190 : メイン制御
194 - 212 : 移動処理
216 - 245 : 移動範囲制限処理
250 - 281 : 弾発射処理
285 - 291 : ダメージ処理
295 - 300 : 移動量テーブル
304 - 311 : 自機パターン
[ボスマップ]
453 - 460 : 動作モード分岐処理
464 - 481 : 初期化
485 - 495 : 登場処理
499 - 516 : 時計まわり処理
520 - 537 : 反時計まわり処理
541 - 560 : 弾発射処理
564 - 570 : 死亡処理
574 - 578 : 消去処理
582 - 588 : ボスパターン

表2 YGCSver.0.30追加コールアドレス一覧

```
NAME     ENTRY ADDRESS
; * * * * * Service call : CHARACTER
ASI_     $30A1     ; Acceleration/slow down init
ASM_     $30A4     ; Acceleration/slow down move
DIRT_    $30A7     ; Direction trace
; * * * * * Service call : GAME
SCOI_    $30C0     ; Score init
SCOS_    $30C3     ; Score set
SCOD_    $30C6     ; Score display
```

注)SYSがついているものは，基本的にシステムが内部で利用するためのシステムコールです。使用する場合には十分注意してください。

表3 YGCS ver.0.30 追加コール

●ASI_

```
in    : H=X start speed
        L=Y start speed
out   : non
break : AF,HL
```

加減速移動の初期速度を定義します。

●ASM_

```
in    : H=X speed
        L=Y speed
out   : non
break : AF,HL
```

指定された加速度で，加減速移動をします。

●DIRT_

```
in    : A=Trace speed
        H=Target X position
        L=Target Y position
out   : non
break : AF,HL
```

目標を方向追尾します。ただし移動はしません。追尾ミサイルなどに使用します。

●SCOI_

```
in    : A=0/P1 1/P2 2/HI
        H=Score data (top)
        L=  〃
        D=  〃
        E=  〃        (bottom)
        HLDE=BCD data
out   : non
break : AF,DE,HL
```

スコアに初期値をセットします。

●SCOS_

```
in    : A=0/P1 1/P2
        H=Score data (top)
        L=  〃
        D=  〃
        E=  〃        (bottom)
        HLDE=BCD data
out   : non
break : AF,DE,HL
```

スコア値を加算します。

●SCOD_

```
in    : A=0/P1 1/P2 2/HI
        L=X position
        H=Y position
out   : non
break : AF,HL,AF'
```

スコアを指定座標に表示します。

リスト1 YGCS ver.0.30

```
3000 C3 00 31 C3 07 31 C3 28 : DA
3008 31 C3 48 31 C3 54 31 C3 : 78
3010 69 31 C3 6A 31 C3 6E 31 : 5A
3018 C3 72 31 C3 76 31 C3 84 : 17
3020 31 C3 92 31 C3 B6 31 C3 : 24
3028 C6 31 C3 CA 31 C3 ED 31 : 96
3030 C3 F5 31 00 00 00 00 00 : E9
3038 00 00 00 00 00 00 00 00 : 00
3040 C3 FD 31 C3 46 32 C3 4D : 3C
3048 32 C3 76 32 C3 9E 32 00 : 30
3050 00 00 00 00 00 00 00 00 : 00
3058 00 00 00 00 00 00 00 00 : 00
3060 C3 B5 32 C3 C5 32 C3 72 : 99
3068 33 C3 8E 33 C3 B1 33 C3 : 21
3070 C6 33 C3 CA 33 00 00 00 : B9
3078 00 00 00 00 00 00 00 00 : 00
-----------------------------------
SUM: 8B BA 1D D1 29 A5 2E 16 CA7C

3080 C3 1B 34 C3 7D 34 C3 BA : 03
3088 34 C3 E8 34 C3 A6 35 C3 : 74
3090 B5 35 C3 E3 35 C3 0C 36 : CA
3098 C3 5E 36 C3 DA 36 C3 0D : FA
30A0 37 C3 66 37 C3 75 37 C3 : C9
30A8 B9 37 00 00 00 00 00 00 : F0
30B0 00 00 00 00 00 00 00 00 : 00
30B8 00 00 00 00 00 00 00 00 : 00
30C0 C3 DD 37 C3 FB 37 C3 54 : E3
30C8 38 00 00 00 00 00 00 00 : 38
30D0 00 00 00 00 00 00 00 00 : 00
30D8 00 00 00 00 00 00 00 00 : 00
30E0 31 00 2F 39 CF 51 42 49 : 44
30E8 42 39 42 3D 42 41 42 45 : 04
30F0 42 51 C3 51 00 00 00 00 : A7
30F8 00 00 00 00 00 00 00 00 : 00
-----------------------------------
SUM: 0F D2 E6 5E 1E 11 45 65 0AC1

3100 CD 48 31 CD 5B 31 C9 3E : A6
3108 01 06 00 CD 4D 32 3E 0E : 9F
3110 06 20 CD 4D 32 CD FD 31 : 6D
3118 21 00 00 22 2F 39 22 34 : 01
3120 39 2A 38 39 22 36 39 C9 : 2E
3128 3A 32 39 47 CD CA 31 32 : E6
3130 32 39 CD ED 31 32 33 39 : F4
```

```
3138 CD 1B 34 CD 7D 34 CD 72 : D9
3140 33 CD C5 32 CD FD 31 C9 : BB
3148 3A 5C 1F 32 CF 51 3E 28 : 6D
3150 CD 30 20 C9 3A CF 51 CD : 0D
3158 30 20 C9 21 2F 39 11 30 : E3
3160 39 01 9F 18 36 00 ED B0 : C4
3168 C9 C9 22 38 39 C9 22 3A : 4A
3170 39 C9 22 3C 39 C9 D5 E6 : 1D
3178 03 87 87 57 1E 00 21 42 : E9
---------------------------------
SUM: 0F B1 A7 74 71 B7 66 57 4D9B

3180 39 19 D1 C9 D5 26 00 29 : 10
3188 29 29 29 29 11 42 49 19 : 59
3190 D1 C9 CD D0 1F FE 20 20 : 94
3198 14 3A 30 39 B7 20 12 3C : DC
31A0 32 30 39 3A 2F 39 EE 01 : 2C
31A8 32 2F 39 3D C9 AF 32 30 : B1
31B0 39 3A 2F 39 3D C9 C5 3A : E0
31B8 31 39 06 AB 80 A8 EE FF : 30
31C0 07 32 31 39 C1 C9 22 3E : 8D
31C8 39 C9 D9 CD D0 1F B7 CA : 18
31D0 EB 31 2A 3E 39 06 0C BE : 8D
31D8 28 06 23 10 FA AF D9 C9 : AC
31E0 ED 5B 3E 39 ED 52 11 93 : A2
31E8 38 19 7E D9 C9 C5 4F 78 : FD
31F0 A9 A1 C1 C9 C9 3A 33 39 : 43
31F8 47 3A 32 39 C9 21 22 46 : 3E
---------------------------------
SUM: 7D 98 A4 BE 7D EE C1 21 5229

3200 D9 06 00 26 00 0E 19 D9 : 05
3208 11 06 00 19 5D 4D 7D 08 : 5F
3210 7C D6 04 57 D6 04 47 D9 : A7
3218 D6 04 57 08 5F 68 06 1A : 20
3220 1A D9 FE 20 20 06 0A FE : 3F
3228 20 20 01 1A BE 28 09 77 : C1
3230 D9 CD 1E 20 D9 CD F4 1F : 9D
3238 03 13 23 D9 13 2C 10 E0 : 41
3240 24 0D C2 07 32 C9 08 3E : 3B
3248 20 08 C3 53 32 08 78 08 : F8
3250 C3 53 32 D9 06 04 21 42 : 8E
3258 39 11 43 39 C5 1F 30 0A : E4
3260 01 FF 03 08 77 08 ED B0 : 27
3268 18 07 01 00 04 09 EB 09 : 21
3270 EB C1 10 E8 D9 C9 D9 B7 : D6
3278 20 0D 21 21 45 11 41 45 : 4B
---------------------------------
SUM: B6 0C CA 4E 24 CD BD 8F 6D2A

3280 01 00 03 ED B8 D9 C9 21 : 6C
3288 50 41 06 19 C5 54 55 1B : 39
3290 01 19 00 ED B8 11 07 00 : D7
3298 19 C1 10 F0 D9 C9 DD 7E : D7
32A0 04 E6 1F 5F DD 7E 06 0F : D8
32A8 0F 0F 57 E6 E0 B3 5F 7A : C7
32B0 E6 03 57 19 C9 D9 21 42 : 5E
32B8 49 11 43 49 01 FF 07 36 : 23
32C0 00 ED B0 D9 C9 DD E5 D9 : DA
32C8 DD 21 42 49 06 40 DD 7E : 2A
32D0 00 B7 CA FD 32 FA FD 32 : D9
32D8 21 42 39 DD 7E 01 B7 F2 : A1
32E0 E5 32 21 42 3D CD 9E 32 : 54
32E8 DD 7E 00 FE 04 CA 53 33 : AD
32F0 FE 03 CA 29 33 FE 02 CA : F1
32F8 0F 33 C3 08 33 11 20 00 : 71
---------------------------------
SUM: 7A 11 CC F7 BB CE 18 65 CFBB

3300 DD 19 10 CA D9 DD E1 C9 : 30
3308 DD 5E 0C 73 C3 FD 32 C5 : 71
3310 DD 5E 0C DD 56 0D EB 01 : 73
3318 20 00 ED A0 ED A0 EB 09 : 2E
3320 EB ED A0 ED A0 C1 C3 FD : 86
3328 32 C5 DD 5E 0C DD 56 0D : 7E
3330 EB 01 20 00 ED A0 ED A0 : 26
3338 ED A0 EB 09 EB 01 20 00 : 8D
3340 ED A0 ED A0 ED A0 EB 09 : 9B
3348 EB ED A0 ED A0 ED A0 C1 : 53
3350 C3 FD 32 C5 DD 5E 0C DD : DB
3358 56 0D EB 4E 23 46 23 3E : 66
3360 20 91 C5 06 00 ED B0 4F : 68
3368 EB 09 EB C1 10 F4 C1 C3 : 28
3370 FD 32 D9 21 42 39 3E 20 : 02
3378 06 00 77 23 77 23 77 23 : D4
---------------------------------
SUM: AB 8B 47 B9 B9 34 EF 7C 47FE

3380 77 23 77 23 77 23 77 23 : 68
3388 77 23 10 EE D9 C9 4F 21 : AA
3390 42 49 19 11 20 00 7E B7 : 0A
3398 28 06 19 10 F9 B7 3F C9 : 0F
33A0 E5 35 23 23 71 23 36 00 : 2A
33A8 11 0B 00 19 36 00 E1 B7 : 03
33B0 C9 E5 21 42 49 19 0E 00 : 81
33B8 11 20 00 7E B7 20 01 0C : 93
33C0 19 10 F8 79 E1 C9 22 40 : A6
33C8 39 C9 F5 FD E5 C5 D5 E5 : 58
33D0 08 2A 40 39 7E B8 38 3C : 55
33D8 23 0E 00 09 09 7E 23 46 : 2A
33E0 16 00 17 17 17 CB 12 17 : 4F
33E8 CB 12 17 CB 12 5F 08 CD : 05
33F0 8E 33 38 20 E5 FD E1 E1 : BD
33F8 D1 C1 DD 7E 04 82 FD 77 : E7
---------------------------------
SUM: E5 F1 6D 66 6F 6C F3 6A 1901

3400 04 DD 7E 06 83 FD 77 06 : 62
3408 FD 74 10 FD 75 11 FD 71 : 72
3410 18 C3 17 34 E1 D1 C1 FD : 96
3418 E1 F1 C9 D9 2A 34 39 23 : 2E
3420 22 34 39 2A 36 39 ED 5B : 70
3428 34 39 4E 23 46 23 EB B7 : E9
3430 ED 42 30 02 D9 C9 EB 7E : 6C
3438 23 B7 FA 6E 34 46 23 5E : 3D
3440 23 56 23 E5 CD 8E 33 38 : 47
3448 1C E5 DD E1 E1 46 23 4E : 57
3450 23 DD 70 04 DD 36 05 00 : 8C
3458 DD 71 06 DD 36 07 00 22 : 90
3460 36 39 C3 26 34 E1 23 23 : B3
3468 22 36 39 C3 26 34 21 01 : D0
3470 00 22 34 39 2A 38 39 22 : 4C
3478 36 39 C3 26 34 CD BA 34 : 47
---------------------------------
SUM: 2D BE 88 BC 05 A9 E6 A7 ADDB

3480 DD 21 42 49 06 40 DD 7E : 2A
3488 00 B7 CA B1 34 DD 7E 03 : C4
3490 3C DD 77 03 16 00 DD 7E : 04
3498 02 87 CB 12 5F 2A 3A 39 : 62
34A0 19 5E 23 56 EB C5 DD E5 : 62
34A8 CD B9 34 DD E1 C1 CD E8 : EE
34B0 34 11 20 00 DD 19 10 CE : 39
34B8 C9 E9 D9 21 42 49 DD 21 : 35
34C0 43 51 11 20 00 0E 00 06 : D9
34C8 40 7E B7 28 12 23 7E 2B : 7B
34D0 B7 F2 DF 34 DD 75 00 DD : EB
34D8 23 DD 74 00 DD 23 0C 19 : 99
34E0 10 E7 79 32 42 51 D9 C9 : D7
34E8 FD E5 D9 3A 42 51 B7 28 : 67
34F0 07 DD 7E 01 B7 F2 FC 34 : 3C
34F8 D9 FD E1 C9 21 42 3D CD : ED
---------------------------------
SUM: 48 91 6A 15 C2 CE 5C 0D A7C1

3500 9E 32 DD 7E 00 47 4F FE : BF
3508 04 20 0B EB DD 6E 0C DD : 4E
3510 66 0D 46 23 4E EB 16 00 : 2B
3518 C5 41 7E FE 20 C2 2F 35 : C8
3520 23 10 F7 3E 20 91 5F 19 : 91
3528 C1 10 EB D9 FD E1 C9 C1 : FD
3530 21 43 51 3A 42 51 47 5E : 27
3538 23 56 23 E5 D5 FD E1 FD : 31
3540 7E 01 DD A6 01 CA 88 35 : 8A
3548 DD 66 04 DD 6E 06 FD 56 : EB
3550 04 FD 5E 06 7A FD 86 14 : 76
3558 4F 7C DD 86 16 91 DA 88 : 37
3560 35 7C DD 86 14 4F 7A FD : EE
3568 86 16 91 DA 88 35 7B FD : 3C
3570 86 15 4F 7D DD 86 17 91 : 72
3578 DA 88 35 7D DD 86 15 4F : DB
---------------------------------
SUM: BE 68 10 29 D4 10 F6 46 643B

3580 7B FD 86 17 91 D2 8F 35 : 3C
3588 E1 10 AC D9 FD E1 C9 E1 : FE
3590 DD 7E 13 FD 96 12 DD 77 : 67
3598 13 FD 7E 13 DD 96 12 FD : 23
35A0 77 13 D9 FD E1 C9 D5 DD : BC
35A8 7E 0E 87 5F 16 00 19 5E : FF
35B0 23 56 EB D1 E9 D9 DD 7E : 52
35B8 10 E6 3F 5F 16 00 DD 4E : D5
35C0 11 21 AF 38 19 E5 7E 5F : F4
35C8 CD 48 36 DD 74 08 DD 75 : F6
35D0 09 E1 11 F0 FF 19 7E 5F : E0
35D8 CD 48 36 DD 74 0A DD 75 : F8
35E0 0B D9 C9 D9 DD 56 08 DD : 9E
35E8 5E 09 DD 66 04 DD 6E 05 : FE
35F0 19 DD 74 04 DD 75 05 DD : A2
35F8 56 0A DD 5E 0B DD 66 06 : EF
---------------------------------
SUM: 00 40 70 0F C0 92 86 FE 6F00

3600 DD 6E 07 19 DD 74 06 DD : 9F
3608 75 07 D9 C9 D9 DD 7E 10 : 62
3610 E6 3F 5F 16 00 DD 4E 11 : D6
3618 21 AF 38 19 E5 7E 5F CD : B0
3620 48 36 DD 56 04 DD 5E 05 : F5
3628 19 DD 74 04 DD 75 05 E1 : A6
3630 11 F0 FF 19 7E 5F CD 48 : 0B
3638 36 DD 56 06 DD 5E 07 19 : CA
3640 DD 74 06 DD 75 07 D9 C9 : 52
3648 16 00 B7 F2 50 36 16 FF : 5A
3650 79 21 00 00 06 08 29 87 : 58
3658 30 01 19 10 F9 C9 C5 D5 : B6
3660 0E 00 7C DD 96 04 30 05 : 36
3668 EE FF 3C 0E 10 5F 7D DD : 00
3670 96 06 30 0B EE FF 3C 57 : 57
3678 79 EE 10 F6 20 4F 7A 57 : AD
---------------------------------
SUM: A8 CC EB 55 4F 7A A8 C6 02EF

3680 7B FE 08 38 06 CB 3A CB : 8F
3688 3B 18 F5 7A FE 08 38 06 : 06
3690 CB 3A CB 3B 18 F5 7B B7 : 4A
3698 CA BB 36 7A B7 CA CB 36 : B7
36A0 17 17 17 B3 5F 16 00 21 : 8E
36A8 EF 38 19 46 79 E6 10 28 : 1D
36B0 05 78 EE 0F 3C 47 78 B1 : 26
36B8 D1 C1 C9 3E 10 08 7D DD : 0B
36C0 BE 06 30 03 3E 30 08 08 : 75
36C8 D1 C1 C9 AF 08 7C DD BE : 29
36D0 04 30 03 3E 20 08 08 D1 : 76
36D8 C1 C9 D9 DD 7E 00 47 4F : 54
36E0 FE 04 20 09 DD 6E 0C DD : 5F
36E8 66 0D 46 23 4E DD 7E 04 : 89
36F0 FE 21 30 14 80 FE 05 38 : 1E
36F8 0F DD 7E 06 FE 21 30 08 : C7
---------------------------------
SUM: EC 62 CE C0 84 FB B0 9C 3FA6

3700 81 FE 06 38 03 AF D9 C9 : 11
3708 AF 3D 3F D9 C9 D5 E5 0E : 95
3710 00 7C DD 96 04 30 04 0C : 33
3718 EE FF 3C 1E 00 1F CB 1B : 4C
3720 1F CB 1B 57 DD 66 04 DD : 80
3728 6E 05 0D 28 04 19 C3 33 : BB
3730 37 ED 52 DD 74 04 DD 75 : 1D
3738 05 E1 0E 00 7D DD 96 06 : EA
3740 30 04 0C EE FF 3C 1E 00 : 87
3748 1F CB 1B 1F CB 1B 57 DD : 3E
3750 66 06 DD 6E 07 0D 28 04 : F7
3758 19 C3 5E 37 ED 52 DD 74 : 01
3760 06 DD 75 07 D1 C9 DD 74 : 4A
3768 08 DD 36 09 00 DD 75 0A : 80
3770 DD 36 0B 00 C9 C5 D5 EB : 6C
3778 DD 66 08 DD 6E 09 4A 06 : EF
---------------------------------
SUM: 7D 42 06 C0 68 5D B2 4D 5253

3780 00 09 DD 74 08 DD 75 09 : BD
3788 4D 44 DD 66 04 DD 6E 05 : 28
3790 09 DD 74 04 DD 75 05 DD : 92
3798 66 0A DD 6E 0B 4B 06 00 : 17
37A0 09 DD 74 0A DD 75 0B 4D : 0E
37A8 44 DD 66 06 DD 6E 07 09 : E8
37B0 DD 74 06 DD 75 07 D1 C1 : 42
37B8 C9 C5 47 CD 5E 36 DD 96 : A9
37C0 10 E6 3F FE 20 30 0B DD : 6B
37C8 7E 10 80 E6 3F DD 77 10 : 97
37D0 C1 C9 DD 7E 10 90 E6 3F : AA
37D8 DD 77 10 C1 C9 C5 FD E5 : 95
37E0 87 87 4F 06 00 FD 21 C3 : 44
37E8 51 FD 09 FD 74 00 FD 75 : 3A
37F0 01 FD 72 02 FD 73 03 FD : E2
37F8 E1 C1 C9 C5 FD E5 44 4D : A3
---------------------------------
SUM: 95 9F 71 F3 27 51 78 2B C153

3800 21 C3 51 B7 28 03 21 C7 : FF
3808 51 E5 E5 23 23 23 7E 83 : 85
3810 27 77 2B 7E 8A 27 77 2B : 9A
3818 7E 89 27 77 2B 7E 88 27 : FD
3820 77 FD E1 D1 21 CB 51 7E : E1
3828 FD BE 00 20 15 23 7E FD : 8E
3830 BE 01 20 0E 23 7E FD BE : 49
3838 02 20 07 23 7E FD BE 03 : 88
3840 20 00 30 0C 21 CB 51 EB : 84
3848 ED A0 ED A0 ED A0 ED A0 : 34
3850 FD E1 C1 C9 C5 D5 EB 87 : 74
3858 87 4F 06 00 21 C3 51 09 : 1A
3860 06 04 EB CD 1E 20 2C EB : 17
3868 7E CB 3F CB 3F CB 3F CB : 67
3870 3F C6 30 CD F4 1F EB CD : CD
3878 1E 20 2C EB 7E E6 0F C6 : 8E
---------------------------------
```

▶とうとうこのときが……封印……。その矢先に付録ディスクが……。我がX68000よ，その電源を入れる日が早くきますように。　　黒澤　典義(17)群馬県

```
SUM: BD 09 FA B6 9A 27 07 3C 3BB5

3880 30 CD F4 1F 23 10 DB D1 : EF
3888 C1 C9 20 20 20 20 20 20 : 4A
3890 20 20 20 01 09 08 0A 02 : 7E
3898 06 04 05 10 20 40 80 00 : FF
38A0 02 03 04 06 07 08 0A 0B : 33
38A8 0C 0D 0E 0E 0F 0F 0F 10 : 72
38B0 0F 0F 0F 0E 0E 0D 0C 0B : 6D
38B8 0A 08 07 06 04 03 02 00 : 28
38C0 FE FD FC FA F9 F8 F6 F5 : CD
38C8 F4 F3 F2 F2 F1 F1 F1 F0 : 8E
38D0 F1 F1 F1 F2 F2 F3 F4 F5 : 93
38D8 F6 F8 F9 FA FC FD FE 00 : D8
38E0 02 03 04 06 07 08 0A 0B : 33
38E8 0C 0D 0E 0E 0F 0F 0F 00 : 62
38F0 00 00 00 00 00 00 00 00 : 00
38F8 08 05 03 02 02 02 01 00 : 17
-----------------------------------
SUM: 2D CF 4E 66 84 91 9F FE 993B

3900 0B 08 06 05 04 03 03 00 : 28
3908 0D 0A 08 07 06 05 04 00 : 35
3910 0E 0B 09 08 07 06 05 00 : 3C
3918 0E 0C 0A 09 08 07 06 00 : 42
3920 0E 0D 0B 0A 09 08 07 00 : 48
3928 0F 0D 0C 0B 0A 09 08 00 : 4E
3930 00 00 00 00 00 00 00 00 : 00
3938 00 00 00 00 00 00 00 00 : 00
3940 00 00 00 00 00 00 00 00 : 00
3948 00 00 00 00 00 00 00 00 : 00
3950 00 00 00 00 00 00 00 00 : 00
3958 00 00 00 00 00 00 00 00 : 00
3960 00 00 00 00 00 00 00 00 : 00
3968 00 00 00 00 00 00 00 00 : 00
3970 00 00 00 00 00 00 00 00 : 00
3978 00 00 00 00 00 00 00 00 : 00
-----------------------------------
SUM: 51 43 38 32 2C 26 21 00 7329
```

リスト2 SAMPLE

```
8000 3E 0C CD F4 1F CD 00 30 : 27
8008 21 F6 84 CD 15 30 21 08 : D6
8010 85 CD 12 30 21 70 85 CD : 77
8018 27 30 CD 03 30 CD 39 80 : DD
8020 CD 54 80 CD CD 1F 28 10 : 92
8028 CD 21 30 28 F6 CD 7B 80 : 04
8030 CD 8C 80 CD 06 30 18 EB : DF
8038 C9 AF 32 7F 85 3E 00 21 : 0D
8040 00 00 11 00 00 CD C0 30 : CE
8048 3E 02 21 00 00 11 00 10 : 82
8050 CD C0 30 C9 3E 02 CD 1B : AE
8058 30 11 E6 00 19 11 06 00 : 57
8060 0E 19 06 1A CD 24 30 CD : 35
8068 24 30 FE F0 3E 20 38 02 : DA
8070 3E 2E 77 23 10 EE 19 0D : 2A
8078 20 E8 C9 3E 00 21 1E 04 : 52
-----------------------------------
SUM: 06 E1 1E 69 45 D8 CC 5C 903C

8080 CD C6 30 3E 02 21 1E 02 : 44
8088 CD C6 30 C9 AF CD 49 30 : 81
8090 3E 02 CD 1B 30 11 E6 00 : 4F
8098 19 06 1A CD 24 30 CD 24 : 4B
80A0 30 FE F0 3E 20 38 02 3E : F4
80A8 2E 77 23 10 EE C9 21 B4 : 64
80B0 80 C3 8C 30 B8 80 E6 80 : 9D
80B8 DD 36 00 03 DD 36 01 01 : 2B
80C0 21 C7 81 DD 75 0C DD 74 : 18
80C8 0D DD 36 12 01 DD 36 13 : 59
80D0 00 DD 36 14 01 DD 36 16 : 51
80D8 01 DD 36 15 01 DD 36 17 : 54
80E0 01 DD 36 0E 01 C9 DD 7E : 47
80E8 13 B7 FA 99 81 21 C7 81 : 47
80F0 DD 75 0C DD 74 0D CD 0C : 95
80F8 81 CD 29 81 CD 58 81 DD : 7B
-----------------------------------
SUM: 4D 36 6E 8D E3 D8 95 65 9437

8100 7E 04 32 7D 85 DD 7E 06 : 17
8108 32 7E 85 C9 CD 30 30 E6 : 11
8110 0F 87 5F 16 00 21 A7 81 : 54
8118 19 DD 7E 04 86 DD 77 04 : 56
8120 23 DD 7E 06 86 DD 77 06 : 64
8128 C9 DD 7E 04 FE 06 30 07 : 63
8130 3E 06 DD 77 04 18 09 FE : BB
8138 1E 38 05 3E 1D DD 77 04 : 0E
8140 DD 7E 06 FE 0A 30 07 3E : DE
8148 0A DD 77 06 18 09 FE 1E : A1
8150 38 05 3E 1D DD 77 06 C9 : BB
8158 06 08 11 20 00 CD 6C 30 : A8
8160 FE 02 38 34 3E 01 06 08 : B9
8168 11 20 00 CD 69 30 E5 FD : 79
8170 E1 DD 7E 04 FD 77 04 DD : 95
8178 7E 06 FD 77 06 3E 01 06 : 43
-----------------------------------
SUM: B3 4B F1 DC 26 46 5A BD CDEF

8180 08 11 20 00 CD 69 30 E5 : 84
8188 FD E1 DD 7E 04 3C 3C FD : B2
8190 77 04 DD 7E 06 FD 77 06 : 56
8198 C9 21 D0 81 DD 75 0C DD : 76
81A0 74 0D DD 36 13 00 C9 00 : 70
81A8 00 00 FF 00 01 00 00 FF : FF
81B0 00 FF FF FF 01 00 00 01 : FF
81B8 00 01 FF 01 01 00 00 00 : 02
81C0 00 00 00 00 00 00 00 20 : 20
81C8 41 20 3C 57 3E 20 59 20 : CB
81D0 20 7B 20 7B 7B 7B 20 59 : A5
81D8 20 21 DF 81 C3 8C 30 E3 : 03
81E0 81 0C 82 DD 36 00 01 DD : 00
81E8 36 01 02 DD 36 0C 21 DD : 56
81F0 36 12 01 DD 36 13 00 DD : 4C
81F8 36 14 00 DD 36 16 00 DD : 50
-----------------------------------
SUM: 5D 13 44 7A 1E 73 83 B5 5C0C

8200 36 15 00 DD 36 17 00 DD : 52
8208 36 0E 01 C9 DD 7E 13 B7 : 33
8210 FA 31 82 DD 7E 06 3D DD : 28
8218 77 06 CD 89 30 DD 7E 13 : 71
8220 B7 FA 31 82 DD 7E 06 3D : 02
8228 DD 77 06 CD 9B 30 38 01 : 2B
8230 C9 AF DD 77 00 C9 21 3C : F2
8238 82 C3 8C 30 40 82 72 82 : B7
8240 DD 36 00 03 DD 36 01 83 : AD
8248 21 9F 82 DD 75 0C DD 74 : F1
8250 0D DD 36 12 01 DD 36 13 : 59
8258 00 DD 36 14 00 DD 36 16 : 50
8260 02 DD 36 15 00 DD 36 17 : 54
8268 02 DD 36 18 00 DD 36 0E : 4E
8270 01 C9 DD 7E 13 B7 FA 8C : 75
8278 82 DD 7E 03 E6 01 47 DD : EB
-----------------------------------
SUM: 4E 2C A5 B6 C5 DF 96 2E 2B0E

8280 7E 06 80 DD 77 06 CD 9B : C6
8288 30 38 0F C9 CD 65 84 3E : 34
8290 00 21 00 00 11 00 02 CD : 01
8298 C3 30 AF DD 77 00 C9 20 : DF
82A0 49 20 5B 2B 5D 4F 48 4F : 32
82A8 21 AE 82 C3 8C 30 B6 82 : 08
82B0 EC 82 FF 82 1F 83 DD 36 : A4
82B8 00 04 DD 36 01 83 21 79 : 35
82C0 83 DD 75 0C DD 74 0D DD : 1C
82C8 36 12 01 DD 36 13 3F DD : 8B
82D0 36 14 00 DD 36 16 06 DD : 56
82D8 36 15 01 DD 36 17 03 DD : 56
82E0 36 10 30 DD 36 11 0C DD : 83
82E8 36 0E 01 C9 26 10 2E 0D : 7F
82F0 CD 9E 30 DD 7E 06 FE 0D : 07
82F8 20 04 DD 36 0E 02 C9 DD : ED
-----------------------------------
SUM: 45 BB AC 85 3C CD 6E 8E 35D0

8300 7E 13 B7 FA 66 83 CD 3F : 37
8308 83 DD 7E 10 C6 02 E6 3F : DB
8310 DD 77 10 FE 30 20 04 DD : 93
8318 36 0E 03 CD 95 30 C9 DD : 7F
8320 7E 13 B7 FA 66 83 CD 3F : 37
8328 83 DD 7E 10 D6 02 E6 3F : EB
8330 DD 77 10 FE 30 20 04 DD : 93
8338 36 0E 02 CD 95 30 C9 DD : 7E
8340 7E 03 E6 07 20 1F 3E 04 : EF
8348 06 08 11 20 02 CD 69 30 : A7
8350 38 13 E5 FD E1 DD 7E 04 : 6D
8358 C6 03 FD 77 04 DD 7E 06 : A2
8360 C6 03 FD 77 06 C9 CD 81 : 5A
8368 84 3E 00 21 01 00 11 00 : F5
8370 00 CD C3 30 AF DD 77 00 : C3
8378 C9 07 05 20 20 20 55 20 : AA
-----------------------------------
SUM: BD 20 2D 2D CF 16 4D 4F 5C62

8380 20 20 49 20 4F DB 4F 20 : 42
8388 49 3C 76 56 5A 56 76 3E : B5
8390 21 20 5B 57 5D 20 21 20 : B1
8398 20 20 55 20 20 20 21 A4 : BA
83A0 83 C3 8C 30 A8 83 E6 83 : 96
83A8 DD 36 00 01 DD 36 01 81 : A9
83B0 DD 36 0C 6F DD 36 12 01 : B4
83B8 DD 36 13 00 DD 36 14 00 : 4D
83C0 DD 36 16 00 DD 36 15 00 : 51
83C8 DD 36 17 00 DD 36 0E 01 : 4C
83D0 3A 7D 85 67 3A 7E 85 6F : 4F
83D8 CD 98 30 DD 77 10 DD 36 : 0C
83E0 11 0C CD 8F 30 C9 DD 7E : CD
83E8 13 B7 FA F6 83 CD 92 30 : CC
83F0 CD 9B 30 38 01 C9 AF DD : 26
83F8 77 00 C9 21 01 84 C3 8C : 35
-----------------------------------
SUM: ED E0 BC AF 85 73 7A E4 011B

8400 30 05 84 1B 84 DD 36 00 : 6B
8408 03 DD 36 01 00 21 4A 84 : 06
8410 DD 75 0C DD 74 0D DD 36 : CF
8418 0E 01 C9 DD 7E 03 E6 01 : 1D
8420 DD 86 06 DD 77 06 DD 7E : 1E
8428 03 21 4A 84 FE 03 38 13 : 3E
8430 21 53 84 FE 05 38 0C 21 : 60
8438 5C 84 FE 07 38 05 AF DD : AE
8440 77 00 C9 DD 75 0C DD 74 : EF
8448 0D C9 20 20 20 20 2A 20 : A0
8450 20 20 20 20 2A 20 2A 2A : 1E
8458 2A 20 2A 20 20 2A 20 2A : 28
8460 20 2A 20 2A 20 3E 05 06 : FD
8468 27 11 20 03 CD 69 30 38 : F9
8470 0F E5 FD E1 DD 7E 04 FD : 2E
8478 77 04 DD 7E 06 FD 77 06 : 56
-----------------------------------
SUM: 16 03 AE 05 D7 EC 14 73 FCA1

8480 C9 3E 05 06 27 11 20 03 : 6D
8488 CD 69 30 38 68 E5 FD E1 : C9
8490 DD 7E 04 FD 77 04 DD 7E : 32
8498 06 FD 77 06 3E 05 06 27 : F0
84A0 11 20 03 CD 69 30 38 4D : 1F
84A8 E5 FD E1 DD 7E 04 C6 04 : EC
84B0 FD 77 04 DD 7E 06 FD 77 : 4D
84B8 06 3E 05 06 27 11 20 03 : AA
84C0 CD 69 30 38 30 E5 FD E1 : 91
84C8 DD 7E 04 FD 77 04 DD 7E : 32
84D0 06 C6 04 FD 77 06 3E 05 : 8D
84D8 06 27 11 20 03 CD 69 30 : C7
84E0 38 13 E5 FD E1 DD 7E 04 : 6D
84E8 C6 04 FD 77 04 DD 7E 06 : A3
84F0 C6 04 FD 77 06 C9 AE 80 : 3B
84F8 D9 81 36 82 00 00 9E 83 : 33
-----------------------------------
SUM: C5 64 FB 8D DC 89 E4 F5 BD3D

8500 FB 83 00 00 00 00 A8 82 : A8
8508 00 00 00 01 00 00 11 19 : 2B
8510 20 00 02 08 20 01 08 04 : 57
8518 28 00 02 08 20 01 0C 04 : 63
8520 30 00 02 08 20 01 10 04 : 6F
8528 38 00 02 08 20 01 14 04 : 7B
8530 40 00 02 08 20 01 18 04 : 87
8538 60 00 02 08 20 01 1B 04 : AA
8540 68 00 02 08 20 01 17 04 : AE
8548 70 00 02 08 20 01 13 04 : B2
8550 78 00 02 08 20 01 0F 04 : B6
8558 80 00 02 08 20 01 0B 04 : BA
8560 A0 00 08 08 20 01 10 02 : E3
8568 FF FF FF 00 00 00 00 00 : FD
8570 38 39 36 33 32 31 34 37 : A8
8578 5A 58 43 56 00 00 00 00 : 4B
-----------------------------------
SUM: 4C 13 94 E2 92 3C AC FC 2EF6
```

▶私のX68000SUPER-HDは，気温が低いときにやたらエラーを出す(特にSX-WINDOW)。フロッピーも一度で食べてくれない。やはり修理に出したほうがいいだろうか。
高橋　哲哉(19)宮城県

リスト3 SAMPLE.ASM(参考)

```
  1: ; ==============================
  2: ;        YGCS SAMPLE GAME
  3: ;     programmed by Yu-Ri.
  4: ; ==============================
  5: ; Test work structure
  6: ;   0 -    : 1: Player
  7: ;   1 -  8 : 8: Player bullet
  8: ;   9 - 16 : 8: Enemy
  9: ;  17 - 24 : 8: Enemy bullet
 10: ;  25 - 63 :39: etc...
 11: ; ==============================
 12:
 13:         CSEG
 14:
 15:         INCLUDE SOSCALL.EQU
 16:         INCLUDE YGCSCALL.EQU
 17:
 18: ; ==============================
 19: ;        TEST PROGRAM MAIN
 20: ; ==============================
 21: START:
 22:         LD      A,$C
 23:         CALL    _PRINT          ; Text cls
 24:
 25:         CALL    COLD_           ; System init
 26:         LD      HL,CHR_TBL
 27:         CALL    CHRF_           ; CHR table address set
 28:         LD      HL,CHR_GEN
 29:         CALL    GENF_           ; CHR gene table address set
 30:         LD      HL,KEY_TBL
 31:         CALL    KEYF_           ; Key table address set
 32:         CALL    HOT_            ; Screen init & etc...
 33:
 34:         CALL    MAIN_INIT       ; Main program init
 35:         CALL    BG_INIT         ; BG draw
 36: MAIN_LOOP:
 37:         CALL    _BRKEY          ; EXIT check
 38:         JR      Z,EXIT
 39:         CALL    PUSC_           ; Pause check
 40:         JR      Z,MAIN_LOOP
 41:
 42:         CALL    SCORE_DISP      ; Score display
 43:         CALL    BG_SCROLL       ; BG scroll
 44:         CALL    MAIN_           ; Game main
 45:
 46:         JR      MAIN_LOOP
 47: EXIT:
 48:         RET                     ; Return S-OS
 49:
 50: ; ==============================
 51: ;          MAIN INIT
 52: ; ==============================
 53: MAIN_INIT:
 54:         XOR     A
 55:         LD      (BS_FLAG),A
 56:
 57:         LD      A,0
 58:         LD      HL,0000H
 59:         LD      DE,0000H
 60:         CALL    SCOI_           ; P1 score init
 61:
 62:         LD      A,2
 63:         LD      HL,0000H
 64:         LD      DE,1000H
 65:         CALL    SCOI_           ; Hi score set
 66:
 67:         RET
 68:
 69: ; ==============================
 70: ;          BG INIT
 71: ; ==============================
 72: BG_INIT:
 73:         LD      A,2
 74:         CALL    VADR_           ; Vram address get
 75:         LD      DE,32*7+6
 76:         ADD     HL,DE
 77:
 78:         LD      DE,6
 79:         LD      C,25
 80: BI_L0:
 81:         LD      B,26
 82: BI_L1:
 83:         CALL    RNDG_
 84:         CALL    RNDG_           ; Random value get
 85:         CP      $F0
 86:         LD      A,' '
 87:         JR      C,BI_L2
 88:         LD      A,'.'
 89: BI_L2:
 90:         LD      (HL),A
 91:         INC     HL
 92:         DJNZ    BI_L1
 93:
 94:         ADD     HL,DE
 95:         DEC     C
 96:         JR      NZ,BI_L0
 97:
 98:         RET
 99:
100: ; ==============================
101: ;        SCORE DISPLAY
102: ; ==============================
103: SCORE_DISP:
104:         LD      A,0
105:         LD      HL,041EH
106:         CALL    SCOD_           ; P1 score disp
107:
108:         LD      A,2
109:         LD      HL,021EH
110:         CALL    SCOD_           ; Hi score disp
111:
112:         RET
113:
114: ; ==============================
115: ;          BG SCROLL
116: ; ==============================
117: BG_SCROLL:
118:         XOR     A
119:         CALL    VRAMS_          ; BG V scroll
120:
121:         LD      A,2
122:         CALL    VADR_           ; Vram address get
123:         LD      DE,32*7+6
124:         ADD     HL,DE
125:
126:         LD      B,26
127: BS_L0:
128:         CALL    RNDG_
129:         CALL    RNDG_           ; Random value get
130:         CP      $F0
131:         LD      A,' '
132:         JR      C,BS_L1
133:         LD      A,'.'
134: BS_L1:
135:         LD      (HL),A
136:         INC     HL
137:         DJNZ    BS_L0
138:
139:         RET
140:
141: ; ==============================
142: ;           PLAYER
143: ; ==============================
144: PLAYER:
145:         LD      HL,PLAYER_TBL
146:         JP      MODC_
147: PLAYER_TBL:
148:         DW      PL_MODE_0       ; 0 : Player init
149:         DW      PL_MODE_1       ; 1 : Player main
150: ;
151: ;       <<< PLAYER INIT
152: ;
153: PL_MODE_0:
154:         LD      (IX),3          ; CHR size 3x3
155:         LD      (IX+ATR),$01    ; Player mode
156:         LD      HL,PAT_PL       ; CHR pattern set
157:         LD      (IX+PAT),L
158:         LD      (IX+PAT+1),H
159:         LD      (IX+POW),1      ; Hit power set
160:         LD      (IX+HPT),0      ; Hit point set
161:         LD      (IX+SXP),1      ; Hit size box
162:         LD      (IX+EXP),1      ;   "
163:         LD      (IX+SYP),1      ;   "
164:         LD      (IX+EYP),1      ;   "
165:
166:         LD      (IX+MOD),1      ; Next move mode
167:
168:         RET
169: ;
170: ;       <<< PLAYER MAIN
171: ;
172: PL_MODE_1:
173:         LD      A,(IX+HPT)
174:         OR      A
175:         JP      M,PL_HIT        ; Hit ?
176:
177:         LD      HL,PAT_PL       ; Normal pattern set
178:         LD      (IX+PAT),L
179:         LD      (IX+PAT+1),H
180:
181:         CALL    PL_MOVE
182:         CALL    PL_LIMIT
183:         CALL    PL_FIRE
184:
185:         LD      A,(IX+XPS)
186:         LD      (PL_X),A        ; Global work set
187:         LD      A,(IX+YPS)
188:         LD      (PL_Y),A        ;   "
189:
190:         RET
191: ;
192: ;       <<< PLAYER MOVE
193: ;
194: PL_MOVE:
195:         CALL    KEYG_           ; Key read
196:         AND     $F
197:         ADD     A,A
198:         LD      E,A
199:         LD      D,0
200:         LD      HL,PL_JTBL
201:         ADD     HL,DE
202:
203:         LD      A,(IX+XPS)
204:         ADD     A,(HL)          ; X move
205:         LD      (IX+XPS),A
206:         INC     HL
207:
208:         LD      A,(IX+YPS)
```


▶Oh!Xの裏表紙のシャープの広告が変わった。X68000XVIはなくなってしまったのだろうか。　清野　晃一(17)静岡県

```
209:          ADD     A,(HL)          ; Y move
210:          LD      (IX+YPS),A
211:
212:          RET
213: ;
214: ;        <<< PLAYER LIMIT
215: ;
216: PL_LIMIT:
217:          LD      A,(IX+XPS)
218:          CP      6               ; Left limit ?
219:          JR      NC,PL_LL0
220:
221:          LD      A,6
222:          LD      (IX+XPS),A
223:          JR      PL_LL1
224: PL_LL0:
225:          CP      30              ; Right limit ?
226:          JR      C,PL_LL1
227:
228:          LD      A,29
229:          LD      (IX+XPS),A
230: PL_LL1:
231:          LD      A,(IX+YPS)
232:          CP      10              ; Up limit ?
233:          JR      NC,PL_LL2
234:
235:          LD      A,10
236:          LD      (IX+YPS),A
237:          JR      PL_LL3
238: PL_LL2:
239:          CP      30              ; Down limit ?
240:          JR      C,PL_LL3
241:
242:          LD      A,29
243:          LD      (IX+YPS),A
244: PL_LL3:
245:          RET
246:
247: ;
248: ;        <<< PLAYER FIRE
249: ;
250: PL_FIRE:
251:          LD      B,8
252:          LD      DE,1*32
253:          CALL    OBJF_
254:          CP      2               ; Work free >= 2 ?
255:          JR      C,PL_FL0
256:
257:          LD      A,1             ; 'Player bullet' ID code
258:          LD      B,8
259:          LD      DE,1*32
260:          CALL    OBJS_           ; Bullet left set
261:          PUSH    HL
262:          POP     IY
263:          LD      A,(IX+XPS)      ; X position set
264:          LD      (IY+XPS),A
265:          LD      A,(IX+YPS)      ; Y         "
266:          LD      (IY+YPS),A
267:
268:          LD      A,1             ; 'Player bullet' ID code
269:          LD      B,8
270:          LD      DE,1*32
271:          CALL    OBJS_           ; Bullet right set
272:          PUSH    HL
273:          POP     IY
274:          LD      A,(IX+XPS)      ; X position set
275:          INC     A
276:          INC     A
277:          LD      (IY+XPS),A
278:          LD      A,(IX+YPS)      ; Y         "
279:          LD      (IY+YPS),A
280: PL_FL0:
281:          RET
282: ;
283: ;        <<< PLAYER HIT
284: ;
285: PL_HIT:
286:          LD      HL,PAT_PLD      ; Damage pattern set
287:          LD      (IX+PAT),L
288:          LD      (IX+PAT+1),H
289:          LD      (IX+HPT),0      ; Hit point set
290:
291:          RET
292: ;
293: ;        <<< MOVE DATA
294: ;
295: PL_JTBL:
296:          ;       dX_dY:dX_dY:dX_dY:dX_dY
297:          DB      00,00,00,-1,00,01,00,00
298:          DB      -1,00,-1,-1,-1,01,00,00
299:          DB      01,00,01,-1,01,01,00,00
300:          DB      00,00,00,00,00,00,00,00
301: ;
302: ;        <<< PATTERN DATA
303: ;
304: PAT_PL:
305:          DB          ' A '
306:          DB          '<W>'
307:          DB          ' Y '
308: PAT_PLD:
309:          DB          ' { '
310:          DB          '{{{'
311:          DB          ' Y '
312:
313: ; ==============================
314: ;             PLAYER BULLET
315: ; ==============================
```

```
316: PL_BULLET:
317:          LD      HL,PLBUL_TBL
318:          JP      MODC_
319: PLBUL_TBL:
320:          DW      PB_MODE_0       ; 0 : Bullet init
321:          DW      PB_MODE_1       ; 1 : Bullet main
322: ;
323: ;        <<< BULLET INIT
324: ;
325: PB_MODE_0:
326:          LD      (IX),1          ; CHR size 1x1
327:          LD      (IX+ATR),$02    ; Player bullet mode
328:          LD      (IX+PAT),'!'    ; CHR code set
329:          LD      (IX+POW),1      ; Hit power set
330:          LD      (IX+HPT),0      ; Hit point set
331:          LD      (IX+SXP),0      ; Hit size box
332:          LD      (IX+EXP),0      ;      "
333:          LD      (IX+SYP),0      ;      "
334:          LD      (IX+EYP),0      ;      "
335:
336:          LD      (IX+MOD),1      ; Next move mode
337:
338:          RET
339: ;
340: ;        <<< BULLET MAIN
341: ;
342: PB_MODE_1:
343:          LD      A,(IX+HPT)
344:          OR      A
345:          JP      M,PB_CLR        ; Hit ?
346:
347:         ;DEC     (IX+YPS)
348:          LD      A,(IX+YPS)
349:          DEC     A
350:          LD      (IX+YPS),A
351:
352:          CALL    HITC_           ; Hit check
353:
354:          LD      A,(IX+HPT)
355:          OR      A
356:          JP      M,PB_CLR        ; Bullet hit ?
357:
358:         ;DEC     (IX+YPS)
359:          LD      A,(IX+YPS)
360:          DEC     A
361:          LD      (IX+YPS),A
362:
363:          CALL    OUTC_
364:          JR      C,PB_CLR
365:
366:          RET
367: ;
368: ;        <<< BULLET CLEAR
369: ;
370: PB_CLR:
371:          XOR     A
372:          LD      (IX),A
373:
374:          RET
375:
376: ; ==============================
377: ;             ENEMY A
378: ; ==============================
379: ENEMY_A:
380:          LD      HL,ENEMY_A_TBL
381:          JP      MODC_
382: ENEMY_A_TBL:
383:          DW      ENA_MODE_0      ; 0 : Enemy init
384:          DW      ENA_MODE_1      ; 1 : Enemy main
385: ;
386: ;        <<< ENEMY A INIT
387: ;
388: ENA_MODE_0:
389:          LD      (IX),3          ; CHR size 1x1
390:          LD      (IX+ATR),$83    ; Enemy mode
391:          LD      HL,PAT_ENA      ; CHR pattern set
392:          LD      (IX+PAT),L
393:          LD      (IX+PAT+1),H
394:          LD      (IX+POW),1      ; Hit power set
395:          LD      (IX+HPT),0      ; Hit point set
396:          LD      (IX+SXP),0      ; Hit size box
397:          LD      (IX+EXP),2      ;      "
398:          LD      (IX+SYP),0      ;      "
399:          LD      (IX+EYP),2      ;      "
400:          LD      (IX+24),0
401:
402:          LD      (IX+MOD),1      ; Next move mode
403:
404:          RET
405: ;
406: ;        <<< ENEMY A MAIN
407: ;
408: ENA_MODE_1:
409:          LD      A,(IX+HPT)
410:          OR      A
411:          JP      M,ENA_DEAD      ; Dead ?
412:
413:          LD      A,(IX+CNT)      ; Count get
414:          AND     1
415:          LD      B,A
416:          LD      A,(IX+YPS)
417:          ADD     A,B             ; Move
418:          LD      (IX+YPS),A
419:
420:          CALL    OUTC_
421:          JR      C,ENA_CLR
422:
```

▶この度，無事ハードディスクドライブが買えそうです。う，嬉しい……。
木村　亮(20)東京都

```
423:         RET
424: ;
425: ;        <<< ENEMY A DEAD
426: ;
427: ENA_DEAD:
428:         CALL    EXA_SET
429:
430:         LD      A,0
431:         LD      HL,0000H
432:         LD      DE,0200H
433:         CALL    SCOS_           ; P1 score set
434: ;
435: ;        <<< ENEMY A CLEAR
436: ;
437: ENA_CLR:
438:         XOR     A
439:         LD      (IX),A
440:
441:         RET
442: ;
443: ;        <<< PATTERN DATA
444: ;
445: PAT_ENA:
446:         DB      ' I '
447:         DB      '[+]'
448:         DB      'OHO'
449:
450: ; ==============================
451: ;         ENEMY D (BOSS?)
452: ; ==============================
453: ENEMY_D:
454:         LD      HL,ENEMY_D_TBL
455:         JP      MODC_
456: ENEMY_D_TBL:
457:         DW      END_MODE_0      ; 0 : Enemy init
458:         DW      END_MODE_1      ; 1 : Enemy on screen
459:         DW      END_MODE_2      ; 2 : Enemy turn right
460:         DW      END_MODE_3      ; 3 : Enemy turn left
461: ;
462: ;        <<< ENEMY D INIT
463: ;
464: END_MODE_0:
465:         LD      (IX),4          ; CHR size nxn
466:         LD      (IX+ATR),$83    ; Enemy mode
467:         LD      HL,PAT_END      ; CHR pattern set
468:         LD      (IX+PAT),L
469:         LD      (IX+PAT+1),H
470:         LD      (IX+POW),1      ; Hit power set
471:         LD      (IX+HPT),63     ; Hit point set
472:         LD      (IX+SXP),0      ; Hit size box
473:         LD      (IX+EXP),6      ;     "
474:         LD      (IX+SYP),1      ;     "
475:         LD      (IX+EYP),3      ;     "
476:         LD      (IX+DIR),48     ; Move direction set
477:         LD      (IX+SPD),12     ; Move speed set
478:
479:         LD      (IX+MOD),1      ; Next move mode
480:
481:         RET
482: ;
483: ;        <<< ENEMY D ON SCREEN
484: ;
485: END_MODE_1:
486:         LD      H,16
487:         LD      L,13
488:         CALL    OPTM_           ; Move
489:         LD      A,(IX+YPS)
490:         CP      13
491:         JR      NZ,END_M1_RET
492:
493:         LD      (IX+MOD),2      ; Next move mode
494: END_M1_RET:
495:         RET
496: ;
497: ;        <<< ENEMY D TURN RIGHT
498: ;
499: END_MODE_2:
500:         LD      A,(IX+HPT)
501:         OR      A
502:         JP      M,END_DEAD      ; Dead ?
503:
504:         CALL    END_FIRE        ; Bullet fire
505:
506:         LD      A,(IX+DIR)
507:         ADD     A,2
508:         AND     $3F
509:         LD      (IX+DIR),A
510:         CP      48
511:         JR      NZ,END_M2_RET   ; Turn direction change ?
512:
513:         LD      (IX+MOD),3      ; Next move mode
514: END_M2_RET:
515:         CALL    SINM_           ; Move
516:         RET
517: ;
518: ;        <<< ENEMY D TURN LEFT
519: ;
520: END_MODE_3:
521:         LD      A,(IX+HPT)
522:         OR      A
523:         JP      M,END_DEAD      ; Dead ?
524:
525:         CALL    END_FIRE        ; Bullet fire
526:
527:         LD      A,(IX+DIR)
528:         SUB     2
529:         AND     $3F
530:         LD      (IX+DIR),A
531:         CP      48
532:         JR      NZ,END_M3_RET   ; Turn direction change
533:
534:         LD      (IX+MOD),2      ; Next move mode
535: END_M3_RET:
536:         CALL    SINM_           ; Move
537:         RET
538: ;
539: ;        <<< ENEMY D BULLET FIRE
540: ;
541: END_FIRE:
542:         LD      A,(IX+CNT)
543:         AND     7
544:         JR      NZ,END_F_RET    ; Fire ok ?
545:
546:         LD      A,4             ; 'Enemy bullet A' ID code
547:         LD      B,8
548:         LD      DE,17*32
549:         CALL    OBJS_           ; Bullet set
550:         JR      C,END_F_RET
551:         PUSH    HL
552:         POP     IY
553:         LD      A,(IX+XPS)
554:         ADD     A,3
555:         LD      (IY+XPS),A
556:         LD      A,(IX+YPS)
557:         ADD     A,3
558:         LD      (IY+YPS),A
559: END_F_RET:
560:         RET
561: ;
562: ;        <<< ENEMY D DEAD
563: ;
564: END_DEAD:
565:         CALL    EXB_SET
566:
567:         LD      A,0
568:         LD      HL,0001H
569:         LD      DE,0000H
570:         CALL    SCOS_           ; P1 score set
571: ;
572: ;        <<< ENEMY D CLEAR
573: ;
574: END_CLR:
575:         XOR     A
576:         LD      (IX),A
577:
578:         RET
579: ;
580: ;        <<< PATTERN DATA
581: ;
582: PAT_END:
583:         DB      7,5
584:         DB      '   U   '
585:         DB      'I O□O I'
586:         DB      '<vVZVv>'
587:         DB      '! [W] !'
588:         DB      '   U   '
589:
590: ; ==============================
591: ;        ENEMY BULLET A
592: ; ==============================
593: EN_BULLET_A:
594:         LD      HL,ENBLA_TBL
595:         JP      MODC_
596: ENBLA_TBL:
597:         DW      EBA_MODE_0      ; 0 : Bullet A init
598:         DW      EBA_MODE_1      ; 1 : Bullet A main
599: ;
600: ;        <<< BULLET A INIT
601: ;
602: EBA_MODE_0:
603:         LD      (IX),1          ; CHR size 1x1
604:         LD      (IX+ATR),$81    ; Enemy bullet mode
605:         LD      (IX+PAT),'o'    ; CHR code set
606:         LD      (IX+POW),1      ; Hit power set
607:         LD      (IX+HPT),0      ; Hit point set
608:         LD      (IX+SXP),0      ; Hit size box
609:         LD      (IX+EXP),0      ;     "
610:         LD      (IX+SYP),0      ;     "
611:         LD      (IX+EYP),0      ;     "
612:
613:         LD      (IX+MOD),1      ; Next move mode
614:
615:         LD      A,(PL_X)
616:         LD      H,A
617:         LD      A,(PL_Y)
618:         LD      L,A
619:         CALL    DIRS_           ; Player direction search
620:         LD      (IX+DIR),A
621:         LD      (IX+SPD),$C
622:         CALL    SINC_           ; Move difference calc
623:
624:         RET
625: ;
626: ;        <<< BULLET A MAIN
627: ;
628: EBA_MODE_1:
629:         LD      A,(IX+HPT)
630:         OR      A
631:         JP      M,EBA_CLR       ; Hit ?
632:
633:         CALL    MOVE_           ; Move
634:
635:         CALL    OUTC_
636:         JR      C,EBA_CLR
```

▶X68000を里子に出して1カ月。PC-9801が手元にあれど……淋しい。
和田　智(18)長野県

```
637:
638:         RET
639: ;
640: ;        <<< BULLET A CLEAR
641: ;
642: EBA_CLR:
643:         XOR     A
644:         LD      (IX),A
645:
646:         RET
647:
648: ; ==============================
649: ;        EXPLOSION A 3x3
650: ; ==============================
651: EXPLOSION_A:
652:         LD      HL,EXPLA_TBL
653:         JP      MODC_
654: EXPLA_TBL:
655:         DW      EXA_MODE_0      ; 0 : Explosion A init
656:         DW      EXA_MODE_1      ; 1 : Explosion A main
657: ;
658: ;        <<< EXPLOSION A INIT
659: ;
660: EXA_MODE_0:
661:         LD      (IX),3          ; CHR size 1x1
662:         LD      (IX+ATR),$00    ; No hit
663:         LD      HL,PAT_EXA0     ; CHR pattern set
664:         LD      (IX+PAT),L
665:         LD      (IX+PAT+1),H
666:
667:         LD      (IX+MOD),1      ; Next move mode
668:
669:         RET
670: ;
671: ;        <<< EXPLOSION A MAIN
672: ;
673: EXA_MODE_1:
674:         LD      A,(IX+CNT)      ; Count get
675:         AND     1
676:         ADD     A,(IX+YPS)
677:         LD      (IX+YPS),A
678:
679:         LD      A,(IX+CNT)      ; Count get
680:
681:         LD      HL,PAT_EXA0     ; CHR pattern 0 set
682:         CP      3
683:         JR      C,EXA_M1_RET
684:
685:         LD      HL,PAT_EXA1     ; CHR pattern 1 set
686:         CP      5
687:         JR      C,EXA_M1_RET
688:
689:         LD      HL,PAT_EXA2     ; CHR pattern 2 set
690:         CP      7
691:         JR      C,EXA_M1_RET
692:
693:         XOR     A
694:         LD      (IX),A          ; Clear explosion
695:
696:         RET
697: EXA_M1_RET:
698:         LD      (IX+PAT),L
699:         LD      (IX+PAT+1),H
700:
701:         RET
702: ;
703: ;        <<<     PATTERN DATA
704: ;
705: PAT_EXA0:
706:         DB      '   '
707:         DB      ' * '
708:         DB      '   '
709: PAT_EXA1:
710:         DB      ' * '
711:         DB      '***'
712:         DB      ' * '
713: PAT_EXA2:
714:         DB      ' * '
715:         DB      '* *'
716:         DB      ' * '
717:
718: ; ==============================
719: ;        EXPLOSION A SET
720: ; ==============================
721: EXA_SET:
722:         LD      A,5             ; 'Explotion A' ID code
723:         LD      B,39            ; etc attribute
724:         LD      DE,25*32        ;         "
725:         CALL    OBJS_
726:         JR      C,EXA_S_RET
727:         PUSH    HL
728:         POP     IY
729:         LD      A,(IX+XPS)      ; Position set
730:         LD      (IY+XPS),A      ;         "
731:         LD      A,(IX+YPS)      ;         "
732:         LD      (IY+YPS),A      ;         "
733: EXA_S_RET:
734:         RET
735:
736: ; ==============================
737: ;        EXPLOSION B SET
738: ; ==============================
739: EXB_SET:
740:         LD      A,5             ; 'Explotion A' ID code
741:         LD      B,39            ; etc attribute
742:         LD      DE,25*32        ;
743:         CALL    OBJS_
744:         JR      C,EXB_S_RET
745:         PUSH    HL
746:         POP     IY
747:         LD      A,(IX+XPS)      ; Position set
748:         LD      (IY+XPS),A      ;         "
749:         LD      A,(IX+YPS)      ;         "
750:         LD      (IY+YPS),A      ;         "
751:
752:         LD      A,5             ; 'Explotion A' ID code
753:         LD      B,39            ; etc attribute
754:         LD      DE,25*32        ;         "
755:         CALL    OBJS_
756:         JR      C,EXB_S_RET
757:         PUSH    HL
758:         POP     IY
759:         LD      A,(IX+XPS)      ; Position set
760:         ADD     A,4
761:         LD      (IY+XPS),A      ;         "
762:         LD      A,(IX+YPS)      ;         "
763:         LD      (IY+YPS),A      ;         "
764:
765:         LD      A,5             ; 'Explotion A' ID code
766:         LD      B,39            ; etc attribute
767:         LD      DE,25*32        ;         "
768:         CALL    OBJS_
769:         JR      C,EXB_S_RET
770:         PUSH    HL
771:         POP     IY
772:         LD      A,(IX+XPS)      ; Position set
773:         LD      (IY+XPS),A      ;         "
774:         LD      A,(IX+YPS)      ;         "
775:         ADD     A,4
776:         LD      (IY+YPS),A      ;         "
777:
778:         LD      A,5             ; 'Explotion A' ID code
779:         LD      B,39            ; etc attribute
780:         LD      DE,25*32        ;         "
781:         CALL    OBJS_
782:         JR      C,EXB_S_RET
783:         PUSH    HL
784:         POP     IY
785:         LD      A,(IX+XPS)      ; Position set
786:         ADD     A,4
787:         LD      (IY+XPS),A      ;         "
788:         LD      A,(IX+YPS)      ;         "
789:         ADD     A,4
790:         LD      (IY+YPS),A      ;         "
791: EXB_S_RET:
792:         RET
793:
794: ; ==============================
795: ;        CHR PROGRAM TABLE
796: ; ==============================
797: CHR_TBL:
798:         ;       Label              ID  Name
799:         DW      PLAYER          ;  0 : Player
800:         DW      PL_BULLET       ;  1 : Player bullet
801:         DW      ENEMY_A         ;  2 : Enemy A
802:         DW      0               ;  3 :
803:         DW      EN_BULLET_A     ;  4 : Enemy bullet A
804:         DW      EXPLOSION_A     ;  5 : Explosion A
805:         DW      0               ;  6 :
806:         DW      0               ;  7 :
807:         DW      ENEMY_D         ;  8 : Enemy D
808:
809: ; ==============================
810: ;       CHR GENERATE TABLE
811: ; ==============================
812: CHR_GEN:
813:         ;  COUNT:COD:SRCHCNT:OFADR:XP:YP
814:         DW  $0000,$00+01*$100,$0000,17+25*$100 ; Player set
815:
816:         DW  $0020,$02+08*$100,$0120,08+04*$100 ; Enemy A set
817:         DW  $0028,$02+08*$100,$0120,12+04*$100
818:         DW  $0030,$02+08*$100,$0120,16+04*$100
819:         DW  $0038,$02+08*$100,$0120,20+04*$100
820:         DW  $0040,$02+08*$100,$0120,24+04*$100
821:
822:         DW  $0060,$02+08*$100,$0120,27+04*$100 ; Enemy A set
823:         DW  $0068,$02+08*$100,$0120,23+04*$100
824:         DW  $0070,$02+08*$100,$0120,19+04*$100
825:         DW  $0078,$02+08*$100,$0120,15+04*$100
826:         DW  $0080,$02+08*$100,$0120,11+04*$100
827:
828:         DW  $00A0,$08+08*$100,$0120,16+02*$100 ; Enemy D set
829:
830:         DW  $FFFF,$FF+00*$100,$0000,00+00*$100 ; Table end
831:
832: ; ==============================
833: ;           KEY TABLE
834: ; ==============================
835: KEY_TBL:
836:         DB      '89632147'      ; Direction key
837:         DB      'ZXCV'          ; Button key
838:
839: ; ==============================
840: ;          GLOBAL WORK
841: ; ==============================
842: PL_FLAG:        DS      1       ; Player flag 1/stage clear
843: PL_X:           DS      1       ; Player position
844: PL_Y:           DS      1       ;         "
845: BS_FLAG:        DS      1       ; Boss flag 1/destruction
846: ; ==============================
```

▶さようなら決算処理，来年は別の人に取りつくんだぞ……。いや本当に冗談抜きで。
岸村　雅史(25)神奈川県

ANOTHER
CG
WORLD
© KYOKO EGUCHI

あれっ
1

2

3

ダメだこりゃ
もう5年も使ってるからねー
4

もう1コあるから
いいやー
ポイッ
5

ずしん
ずしん
6

つら〜
7

ああなたさまはいったい…?
8

わしゃ
マウス大魔神
じゃー
いま おぬしは、マウスを
ポイ捨てしようと
したな……
9

返すことばがない
10

マウスはなー
11

ボールをはずして
汚れを洗ったり…
12

プラグを きれいに
磨くと、復活する
ことがあるんじゃ…
13

こんなあたりまえのこと
もしないで、すぐポイ
するなんて、パソコン
ユーザーの風上にも
おけない
やつじゃ
ヒェー
どーか
おゆるしを～
14

かつ かつ かつ かつ かつ かつ かつ かつ かつ かつ かつ かつ かつ かつ かつ
15

マウスさま….
えーどこか
おカユイところは
ございませんで
しょうか….
23 APRIL '94
なんだかウンチクマンガみたいになってしもた…
16

鬼婆白町から出られない

Takazawa Kyoko
高沢 恭子

最近はあまり手紙を書かない，なんてひとも多いようですが，それでも，送られてくるさまざまな知らせを受け取るときは，ちょっとどきどきしますよね。今回はそのどきどきを運ぶひとのお話です。

晴れた日も雨の日も，彼女のもとに彼からの手紙が届いた。風の日も雪の日も，それは欠かされることがなかった。そして，彼と彼女は結ばれた。もちろん，郵便やさんと彼女である。外国の小話にそんなのがあったっけ。

書留で春がくる

早春の郵便やさんは沈丁花(じんちょうげ)の香りにのって,ほんとに毎日のように玄関先まで通ってくれた。そして，トオルの大学入試戦線も終わった。

書留や速達，電子郵便の利用が欠かせない大学受験ではあるけれど，ひとりの受験生のために，こんなにたくさんの郵便やさんが入れ代わり立ち代わり，いちいちドアをノックしては通知を手渡してくれるなんて，まったくありがたいやら，申しわけないやら。郵便やさんがいなければやってこない春でもあった。

願書を提出するときには，1つの受験に関して，これほど多くの郵便物が行き来するとは思いおよばなかった。それが複数の受験となれば，数は倍増する。トオルの場合，本命にしていた大学だけでも3学部を受験した。それがすべて2次試験まであったので，郵便やさんにいっそうのご足労をかけた。

受験票の到着，1次の結果，2次の結果，入学手続きの書類。これらはみんな書留の郵便だから，これだけ3倍しても12回の宅配になる。ほかにもいくつかの受験をしたが，ざっと数えたら約1カ月のあいだに，およそ延べ30人の郵便やさんのお世話になっていた。

郵便やさんにこだわるのは，夏，冬の休暇になると，暑中見舞い，年賀状の配達要員として，高校生のトオルのところにいくつかの管轄の郵便局から，郵便配達のアルバイトに協力してもらえないかという依頼がよく届いていたからだ。宅配業とはちがって，新聞や郵便を配る人はずいぶん前から不足しているらしい。社会参加への経験やいくらかの奉仕の意味でも，協力できたらいいにはちがいないが，なかなか実行させるような勇気がなかった。

その本人のもとへ，いかに特別料金の郵便物とはいえ，たくさんの手がパーソナルな配達をしてくれている。なんとぜいたくでワガママなんだろうと，また申しわけなくなる。

それでも，だいじな知らせには足を運んでもらう意義も大きい。ところが，じっさいには2次試験の日と重なることになったとか，おっくうになったなどで，放棄してしまった試験もいくつかある。受験していない試験の結果は不要なのに，郵便やさんは悪天候にもめげず玄関のチャイムを押してくれる。これは郵便やさんのムダづかいだ。郵便やさんごめんなさい。

バイクだっていい

テレビ番組「ポンキッキーズ」のレギュラー，ポストマン・パット氏は，いつも自分の郵便車に「ブチ猫」ちゃんを乗せて，村じゅうに郵便を配っている。あんまり急ぐようすもなく，行く先々で村の人たちのしごとを手伝ったり，よけいなお世話をしたりの毎日だ。

わが家のホンニャアはやっとこのごろチャイムの音が怖くなくなったところだったのに，この1カ月の郵便やさんの来襲で，すっかり元にもどってしまった。ホンニャアは，申しわけないが郵便やさんをキライになったらしいので，ブチ猫ちゃんのように郵便やさんの相棒になることはけっしてないだろう。

郵便やさんのいでたちは，ほんとのところちょっと重苦しいものがある。ふだんの服装も晴雨両用，万能タイプといったふうだが，悪天候となるともっと重装備だ。雪の日に玄関に訪れてくれたときは，私だって，感謝しながらも，沿岸警備隊みたいだなと思ったものだ。ホンニャアが敬遠するのも無理ない点もある。

郵便やさんという呼び名には，明るいメルヘンの響きがある。雨の日は似あわない。晴れた日に蝶々に追われながら，みんなの幸せをねがってよい便りだけを配っている人。それも，徒歩か，乗り物はせいぜい自転車どまりだ。

じっさいはそんなわけにはいかない現実的なしごとで，つらい便りも運ばなくてはならない。人口急増地区の部類に入るこの町などは，配達の量もさぞ多いことだろうし，晴雨兼用の上着に，もちろん乗り物はバイクでなければダメだ。メルヘンの世界にはないダイレクトメールも数知れず，こうなるともう，郵便やさんなんて甘い呼び名は似合わなくなる。

郵便やさんが正しい呼び名ではないのはみんな知っている。花屋さん，くつ屋さんというのとはいっしょにできない。正しくは郵政省所属の○○郵便局集配課員，あるいは郵便配達員というそうだが，正しい言葉とはずいぶん味気ないものだ。

おまわりさん，大工さん，板前さんといった呼び名には，このごろはやりのカタカナ職業にはないなつかしさがある。あたたかさや笑顔，人の交流が思い浮かぶ。バイクで電子郵便やダイレクトメールを運んでも，郵便局集配課員より，やっぱり郵便やさんがいい。

索引コードの町

公団マンションの集合郵便受けに，郵便やさんが郵便物を分け入れていた。あらかじめ順序を考えて，よく準備をしてからの配達にちがいないが，1つの棟に百以上の世帯があるだろうから，まちがえないように入れていくのはたいへんだ。

郵便やさんにとって，同じ数の家に配るとしたら，一戸建ての多い住宅街で一軒ごとに郵便受けの前に立ち止まって郵便物を入れるより，マンションなどの集合住宅の郵便受けに入れるほうが能率的ではある。ただし，1つ隣の箱は隣家なのだから，誤配の可能性もふえるかもしれない。

ふと気づいたのだが，郵便やさんの仕分けや配達の作業は，文字が読めなくてもいいのではないだろうか。もちろん文字の読めない郵便やさんがいるとは思わないが，通常の郵便物を配る場合は，文字は見るだけで，その正しい読みや意味は，知っていても知らなくてもかまわないのだ。

郵便物があて名の場所にきちんと届くのを確実に助けているのは，住居表示である数字，すなわちコード番号だ。「5－7－8－503」といったような数字が，あて名の場所を限定してくれる。

すこし前に購入した「JIS第1・2水準漢字索引辞書・漢ぺき君」が，このところ机の上にのったままになっている。

「漢ぺき君」は皮ふ科のお医者さんである高田任康氏が10年がかりで完成された，独創的な発想による，ワープロなどの日本語入力のための漢字辞書である。昨年から新聞などで何回か紹介され，興味をもって買ったものだ。

高田さんの最大のねらいは，おもに単漢字変換のときに，多すぎる候補のなかから選び出す手間と時間を最小限にしようとすることにある。それから，読めない文字を変換するとき，その字がどこにあるかをすぐ知ることができること。また，JIS第2水準までに入っていないような特殊な文字が，その辞書ソフトにあるかどうかを，すぐにわかるようにすることである。

高田さんは，日本語入力するときの単漢字変換の作業は，その漢字がどこにあるかをさがし出すこと，つまり漢和辞典の索引を見ているのと同じであるという点に着目して，この「索引辞書」の作成にあたられたようだ。

いわれてみればその通り，その字はなんと読むのか，本来どんな意味があるかを知るためには，まず辞典のどこにあるかをさがすことが先決だが，ワープロの漢字変換は，どこにあるかということがわかった段階で，しごとが終了するのである。

300年前に中国でつくられた康熙字典の配列がいまだに受け継がれている漢和辞典には，不便な点も多いと感じた高田さんは，画期的な方法を編み出した。それは，漢字を構成要素にしたがって分解し，きめられた順序で部分の読みをおこなう。そのあと，それぞれの読みの最初の1文字だけを順に3つつなぎ合わせる。4つ以上に分解できる文字でも，3つまででよい。

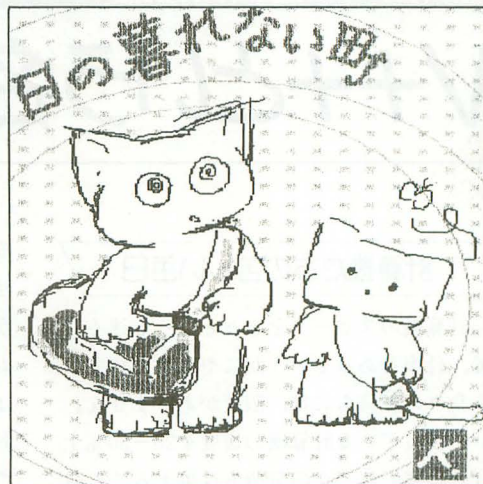
たとえば本書の例を拝借すると，「賢」の場合，「臣」「又」「貝」に分かれ，それぞれ「じん」「また」「かい」になる。これらのはじめの1字をつなぎ合わせると，「じまか」となり，この書の50音順の索引「じまか」の項を見ると，区点コードとシフトJISコードが表示されている。

この字はどこにあるか。まさに郵便やさんの住居さがしと同じだ。コードさえわかれば，文字のある場所がわかる。はじめからコードのついている郵便物は，その場所にもっていけば配達が終わる。そのとき，あて名と同じ文字があればいい。

大発明を生かすには

「漢ぺき君」で，すべての漢字を新しい考えときまりのもとに配列しなおした高田さんのしごとは，新しい学問でもあり，たいへんな偉業と思う。

ただ，一見親しみやすそうな「漢ぺき君」だが，使いこなすには少々努力がいる。この書できめられた方法を身につけることと，漢字についてそれなりの知識も必要だ。そのうえで，これならイケルと思ったら自分の機種に合わせた電子版「漢ぺき君」を購入すればいい。

ちかごろは漢字を見るたびに，「漢ぺき君」流の読みかたをしてみる。「韻」という字は，「立」「日」「口」と3つまで分けて，「たひく」。「街」なら，「彳」と「土」が2つで，「ぎつつ」。

illustration：Kyoko Takazawa

入学準備のためにトオルと買い物に出かけたら，地下鉄の車内に「蒟蒻」(こんにゃく)と書いたお菓子の広告があった。

「蒟」はどう読めばいいだろう。「草かんむり」と「立」と「句」で「くたく」かな，あるいは「勹」(つつみがまえ)まで分けて「くたつ」だろうか。「蒻」はどうなるか。「草かんむり」のつぎは「弓」にして「冫」(にすい)なら，「くゆに」だ。

なんだか，いままでのワープロ対応辞典で見るほうがラクな気もする。

どこかの駅についてドアがあいた。

たくさんの人が乗り降りするなか，騒音にまじって女声のアナウンスがホームに流れた。「オニババシロマチ，ご注意ください」。2度くりかえした。

「いま，鬼婆白町，ご注意くださいっていわなかった？」と，トオルに聞いた。

「トビラがしまります，ご注意くださいっていったんだよ」

つぎの駅についた。やっぱりまた「オニババシロマチ」とたしかにいった。トオルが笑いをこらえている。

「お母さん，疲れてない？」

下車駅について電車を降りた。ホームいっぱいに，「オニババシロマチ，ご注意ください」と響いた。どこまでいっても「鬼婆白町」だ。

山のような郵便を，配っても，配っても減らない郵便やさん。空はねむたそうに白くて，歩いても歩いても日が暮れないヘンな町。そんな町があったら，そこはきっと「鬼婆白町」だから，郵便やさん，気をつけてネ。

V++という名前のプロセッサアーキテ

計算機たちの苦しい毎日

計算機のスピードをもっともっと速くして，仕事をあっという間にやらせたいという我々の気持ちには，限度がありません。いま使っている計算機の速度を20～30%上げた程度では，この気持ちは抑えることはできないでしょう。たとえ，2倍3倍速くなったとしても，またすぐにもっともっと速くしたいという気持ちに陥るであろうことは目に見えています。

ここでひとつの疑問が湧き起こります。なぜ，我々は計算機のスピードに対する期待感を膨大させ続けるのでしょうか？

もちろん，そのおもな理由は，計算機にやらせたいと我々が思っている仕事そのものがどんどん複雑化し，また量的にも，とどまるところなく増大しているからであるといえます。

たとえば昔は，計算機とは，その名前に端的に表れているように，単に数字を入力すると何か計算をして結果を出力してくれるような「計算をする機械」にほかなりませんでした。ところがいまではどうでしょう。グラフィックスだ，画像だ，音声だ，ムービーだ，仮想現実だ，などと，計算ではなく，まるでまったく別の機械に対して向けるべき要求を計算機に対してつきつけているのです。

我々がさせたいとする仕事自体が重くなってきたというこの客観的な理由以外にも，これでもかこれでもかと計算機に対してスピードを要求してしまう心理的な理由もあるようです。それは，「計算機はあっと驚くほど速くて当たり前である」「人をいらいら待たせるような計算機はクズである」とでもいうような思いでしょう。これは，確かに当然ともいえることです。

ただし，先に書いたように，計算機に対する要求は(半分無意識的に)たえず増大していますし，さらに「慣れ」という要因もあります。最初は速いなあと感じても，すぐにその速さに慣れてしまい，何とも思わなくなってしまうというのが「慣れ」です。

illustration：Haruhisa Yamada

最初は「あっと驚くほど」速くても，あっという間に「人をいらいら待たせる」計算機へと，時間とともに自動的に変わってしまうのです。

このように考えてみると，計算機に対する要求は日増しに大きくなる一方であり，さらに，速いか遅いかというものさしの目盛り自体もぐんぐんと伸びてくる毎日であるわけでして，そういう意味で，計算機たちは，きわめて苦しい毎日を送っているといえましょう。

粒の大きさについて

そういう計算機たちに救いの手を，ということで僕たちのグループは日夜がんばっています。手段は並列処理です。計算機を人間にたとえて説明するならば，あるひとりの人をよく教育したり訓練したりして仕事をうまく迅速にこなせるようにしようとすることよりも，もっと多くの人が協力し合って仕事をするようにして，全体としてすばやく仕事をこなせるようにしようとしているのです。

並列処理とひとことでいっても，それにはまた，さまざまなやりかたがあります。たとえば，計算機を複数台つなげる，プロセッサを複数個つなげて実行させる，プロセッサのなかで複数の命令を実行する，などと考えることができます。そのなかで我々は，ほかの並列処理を意識しながらも，最後に書いたやりかた「プロセッサのなかで複数の命令を同時に実行する」ということに焦点をしぼっています。

このような並列処理は，「細粒度並列処理」や「命令レベル並列処理」などと呼ばれます。プログラムとプログラムを並列に実行する，あるいは関数と関数を並列に実行するという場合にくらべて並列処理の単位が命令1つ(あるいはそれより小さい)ということですから，粒が細かいという意味で細粒度というわけです。

どの粒度に注目して並列度を得るかという選択があるわけなのですが，うまいこと

クチャ

やれば，各階層(各粒度)で得られた並列度の積が全体の並列度になるということが重要です。

たとえば，あるプログラムを実行するときに，プログラムを関数のかたまりに分解し，その関数ごとにプロセッサを割り当てて並列実行した結果，並列度が4だけ得られたとします。つまり4倍速くなったわけです。さらに，各プロセッサの内部で命令単位に並列実行を行い，並列度が3だけ得られたとします(これはひとつの関数の実行時間がさらに3倍速くなったことを意味します)。

このようにすれば，全体として得られた並列度は3×4で12となるわけです。つまり，ひとつのプログラムをひとつのプロセッサで逐次的に実行していたときにくらべて12倍の速度向上が達成できたわけです。

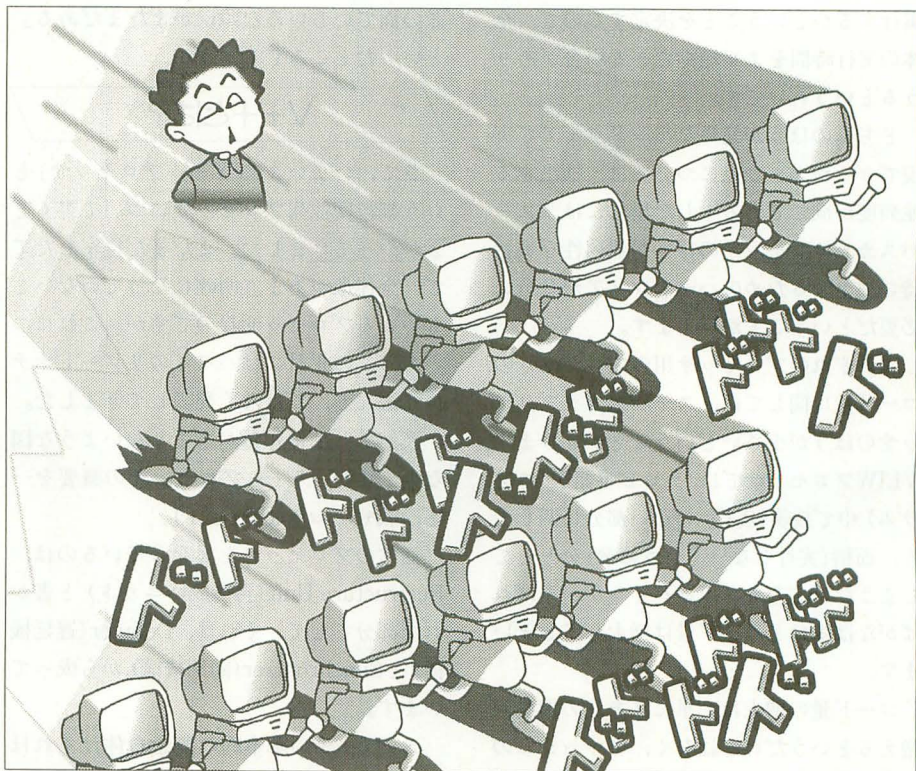
このようなわけで，並列性を抽出して速度を向上させるためには，全体の並列性の抽出を考慮しながら，全体の速度向上に直接貢献するように(掛け算で効果が上がるように) することが大切なのです。

スーパスカラ方式対VLIW方式

細粒度並列プロセッサとして，次の2つの代表選手を挙げることができます。

・スーパスカラプロセッサ
・VLIWプロセッサ

スーパスカラ(superscalar) プロセッサというのは，従来の逐次型プロセッサが実行するのと同様な逐次的なプログラムを実行しますが，プロセッサ内部で命令を順に検査し，並列に実行可能であるとわかると次々に実行を開始していくというタイプのプロセッサです(もちろん,プロセッサには並列実行できるだけの余分なユニットが装備されています)。

一方，VLIW(Very Long Instruction Word)プロセッサというのは，命令Aと命令Bを並列に実行し，次に命令Cと命令Dを並列に実行させる，というような並列実行の形態をあらかじめコンパイラが決定し

ておき，その形態のとおりにプロセッサがメモリから複数の命令をとってきて順に並列実行していくというものです。

たとえていうならば，新聞のテレビ欄のように，どの時間にどの局(ユニット)でどの番組(命令) が放映(実行) されるのかをあらかじめ決めておくのがVLIW方式であり，これに対し，どの局(ユニット)でどの時間に放映(実行)されるかはわからないが順番だけが1列になってあらかじめ決まっており，放映(実行)時になってから，空いている局(ユニット) に次の番組(命令) を渡して放映(実行)していくというのが，スーパスカラ方式といえます。

では，それぞれを比較してみましょう。

まず，コードの互換性という点を考えます。互換性が保たれるならば，いままでの計算機システム中でプロセッサだけを新しい細粒度並列型の高速版に代えて，コンパイラなどはまったく変更せずに，従来のオブジェクトコードが実行できることになります。これについては，スーパスカラプロセッサでは互換性をかなり保つことができます。一方，VLIWプロセッサでは，かなり複雑なコンパイラが必要となり，プログラムもすべてコンパイルし直さないと実行できません。

実行速度に関しては，一概にどちらが速いというのは難しいのですが，VLIWプロセッサでは，実行前にコンパイラがいくらでも解析して並列性を抽出することができるので，実行時にあわてて並列実行を行うスーパスカラプロセッサよりも並列性が十分抽出できる可能性があります。さらに，実行時に並列性を検出する処理が不要なので，その処理にかかる時間の分，より優位になる場合も考えられます。

一方，スーパスカラプロセッサの特長としては，命令実行時間の動的なタイミング変動に強いということが挙げられます。ある命令の実行時間が予測できず，実行時の条件によって変わる，ということはしばしば起こることです。スーパスカラプロセッサでは実行時にどの命令をどのユニットで

V++という名前のプロセッサアーキテクチャ

実行するかということを決定するので，全体の実行時間をより短縮できる場合がありうるというわけです。

どちらのほうが簡単なハードウェアで実現できるかという点については，抽出する並列度が同じであるとした場合には，スーパスカラプロセッサのほうが並列性を実行時に抽出するためのハードウェアが余分に必要だということがいえます。

それぞれのプロセッサ用のプログラムのコード量に関しては，スーパスカラプロセッサのほうが小さいということができます。VLIWプロセッサでは，テレビ番組(プログラム)中で放映(実行)のない部分に関しても，面積(実行しないという命令)をとってしまうだけでなく，コンパイラががんばればがんばるほどコード量は増大してしまいます。

コード量の増大は，単にメモリの消費が増えるというだけではなく，キャッシュのヒット率(高速なメモリ上に命令が存在する確率)という，実行速度に直結する部分に影響が出てくるので，案外と重要な問題なのです。

むりやりに結論づけるとすると，「VLIWプロセッサのほうが最終的には速くなりそうだが，コンパイラ，コード量，互換性などの面でいろいろな障壁がまだまだある」といったところでしょうか。

図1　V++の概要図

V++とは?

さて，いよいよ，「V++プロセッサ」という細粒度並列プロセッサについて書くときがやってきました。まだまだ提案したてで，全体像があまり明確になっていないともいえるプロセッサなのですが，これについて，この正月にはハワイのリゾートホテルで遊んで，ではなく発表してきました。

漠然として情報量があまりないような図なのですが，V++プロセッサの概要を一応，載せておきます(図1)。

通常のプロセッサと異なっているのは，Restructure Unit(再構成ユニット)と書かれた部分でして，それは，Delayer(遅延機構)とSynchronizer(同期機構)から成っています。

このV++の基本的な概念自体はそれほど難しくないのですが，いちばん理解してほしいところを理解してもらうのは案外と難しいかもしれません。別の研究者が，V++について理解しているつもりで話していても，聞いているうちに，実はその人はV++のごく一部しか理解できていないのだとわかることはよくあります。それどころか，共同で研究している人でさえも，何カ月かたってからやっとV++の面白さというか，美しさというものがわかったというほどです。

基本的なV++の枠組みは次のようにして考えました。「ハードウェアがシンプルでコンパイラによって十分に並列性を抽出できるVLIWプロセッサをまず想定し，そこに，スーパスカラプロセッサのもつ特長である動的に並列性を抽出する機能を導入して，両者の特長を活かそう」ということです。

結果として，ハードウェアが簡単である，あるいは，実行時の処理が簡単なので高速化がしやすいなどの，VLIWプロセッサのもつ特長に加えて，次のような特長が得られました。

・命令の実行時間が予測と異なった場合でもその影響を抑えることが可能である。

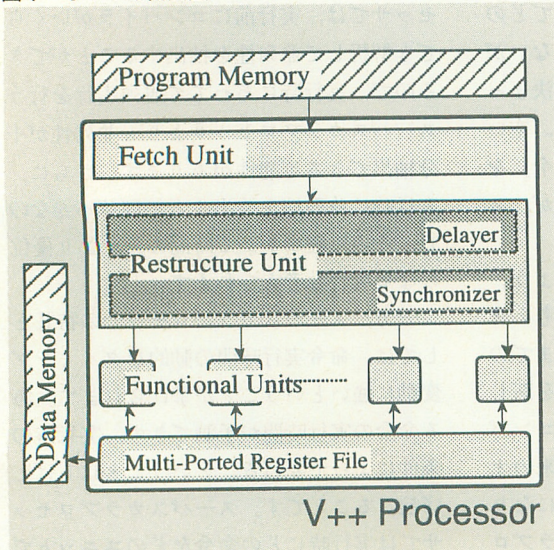
・命令コードの量の増加を抑えることができる。

アーキテクチャの世界

今回は，僕たちが手掛けているV++という名前のプロセッサについて，そのものの説明自体はあまりできませんでしたが，だいたいの位置づけはできたと思います。

アーキテクチャとは，理論だけではうまくいかない難しい世界です。というのは，現実の素子を相手にしているので，そこに発生するいろいろな物理的事実を踏まえる必要があるという面もありますし，現実に存在しているソフトウェア資産をどう活かすかという面もあります。また，ハードウェアやソフトウェアにかかるコスト，つまりお金を考えなければならないということもあるからです。

さらに，何がよいアーキテクチャかという面に関しては，統一されたひとつのものさしというものは存在しません。いろいろな思想というか，哲学というか，場合によっては宣伝文句とでもいうべきものが，そのときそのときで流行しやすい領域であるともいえるのです。

僕にとっては，あの懐かしいMZ-80KにおいてカセットによってBASICが，SP-5001，5002や5010などになって速くなったとか，CPUが上位のものにバージョンアップしてゲームをしたら，速すぎてできなくなったとか，Macintoshのウィンドウを開くスピードが速すぎて見えなくなったとか，挙げていくときりがありませんが，そのような瞬間瞬間にわき上がるあの忘れがたい気持ちと，速いプロセッサを作りたいという現在の意欲は，やはり僕の根底でいつもつながっているのです。

創刊12周年記念

愛読者特大プレゼント

読者の皆様の深い愛情にささえられて，はや12年。長いあいだにはいろいろなことがありましたが，Oh!Xも読者とともにこの歳月を歩んできました。いままでのありがとうとこれからのよろしくの気持ちを込めて，特大プレゼントです。最新ソフトから，手に入りにくい懐かしのソフトや非売品グッズまで，お馴染みのメーカーさんからいろいろご提供いただきました。たくさんのご応募をお待ちしています。

シャープ ☎03(3260)1161

1 Easydraw SX-68K 5名
X68000用 3.5/5"2HD版
19,800円(税別)

2 Communication SX-68K 5名
X68000用 3.5/5"2HD版
19,800円(税別)

3 SX-WINDOW デスクアクセサリ集 5名
X68000用 3.5/5"2HD版
14,800円(税別)

4 CYBERNOTE PRO-68K 5名
X68000用 3.5/5"2HD版
19,800円(税別)

EAビクター ☎03(5410)3111

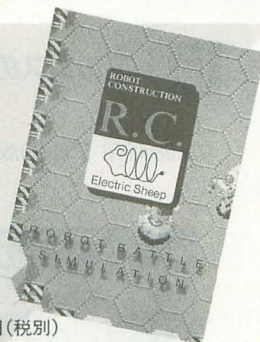
5 コットン 3名
X68000用 5"2HD版
9,800円(税別)

6 コットン テレホンカード 10名
非売品

SPS ☎0245(45)5777

7 ネメシス'90改 3名
X68000用 5"2HD版
8,800円(税別)

エレクトリックシープ ☎052(775)0530

8 ロボットコンストラクションR.C. 2名
X68000用 5"2HD版
8,800円(税別)

9 R.C.ロボット集＋α Vol.2 3名
X68000用 5"2HD版
1,800円(税込)

カプコン

10 ファイナルファイト

1名

X68000用 5"2HD版

9,800円(税別)

11 ストリートファイターII ダッシュ

X68000用 5"2HD版

12,800円(税別)

光栄 ☎045(561)4666

12 ヨーロッパ戦線

1名

X68000用 5"2HD版

12,800円(税別)

13 ロイヤルブラッド

1名

X68000用 5"2HD版

7,800円(税別)

14 リーディングカンパニー

1名

X68000用 5"2HD版

12,800円(税別)

15 太閤立志伝ハンドブック

2名

1,860円(税込)

16 三國志IIIハンドブック

2名

1,860円(税込)

17 伊忍道ハンドブック

2名

1,860円(税込)

電波新聞社 ☎03(3445)6111

18 エキサイティングアワー/出世大相撲

X68000用 5"2HD版 3名

5,300円(税別)

19 アルゴスの戦士

X68000用 5"2HD版 3名

5,300円(税別)

ビング ☎03(3492)1079

20 スーパーリアル麻雀PIV

ポスター 5名

※ポストカードⒶの絵柄です。 非売品

21

Ⓐ Ⓑ

スーパーリアル麻雀PIV

ポストカード(2枚組) 10名

非売品

ファミリーソフト ☎03(3924)5435

22 あすか120% BURNING FEST

テレホンカード

5名

非売品

ブラザー工業 ☎052(824)2493

23 宝魔ハンターライム 第1話～第6話セット

X68000用 5"2HD版 3名

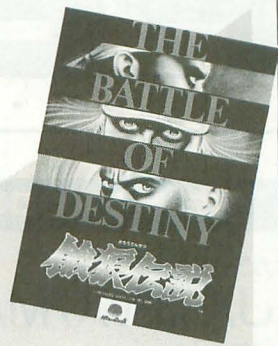
各1,500円(税込)

魔法株式会社 ☎078(261)2790

24 餓狼伝説

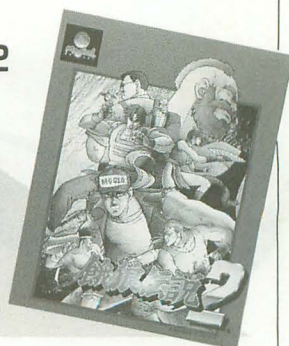
3名

X68000用 5"2HD版

8,800円(税別)

25 餓狼伝説2

2名

X68000用 5"2HD版

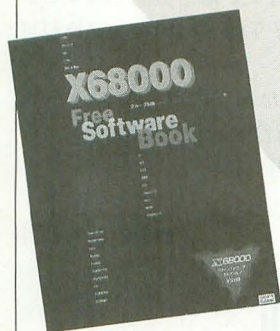
9,800円(税別)

ソフトバンク ☎03(5642)8101

26 Free Software Book

5名

2,900円(税込)

27 040turboステッカー

X68040 turbo X68040 turbo
X68040turbo

20名

非売品

プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1994年6月18日の到着分までとします。当選者の発表は1994年8月号で行います。また，雑誌公正競争規約の定めにより，当選された方はこの号のほかの懸賞には当選できない場合がありますので，ご了承ください。

Oh!X編集部

28 福袋 X1/X1turbo用

5名

29 福袋 MZ-2500用

2名

Oh!X編集室の魔境から発掘されたソフトなどが入っています。なんと未使用。袋詰め作業は懐かしさに涙しつつ行われました。

30 Oh!X特製シャープペンシル

非売品 10名

Oh!Xのロゴ入りです。軸の色はワインレッドとグレイで消しゴムつき。2本セットでプレゼントします。

4月号プレゼント当選者

1マッドストーカーX68 (宮城県)伊東浩治 (埼玉県)加藤洋介 (和歌山県)加納伸吾 2キーパー (宮城県)阿部 嘉 (東京都)山下佳寿 (鹿児島県)前田多門 3デジタルアートコレクション Avol.1 (青森県)水上秀博 (千葉県)上田幸治 田村日出刀 (富山県)平 好博 (滋賀県)若林正浩 Bvol.3 (東京都)榎本喜世史 広野 徹 (静岡県)青島一高 望月利修 (鹿児島県)福留敏和 Cvol.5 (群馬県)皆川朋久 (茨城県)渡辺昭文 (新潟県)金子直史 (愛知県)岡本壮紀 (香川県)伊藤浩克 Dvol.7 (千葉県)三谷夏樹 (神奈川県)安沢光男 (愛知県)小川克仁 (大阪府)村上忠史 (兵庫県)徳富優一 4ACPSファイター (宮城県)千葉浩貴 (福島県)国井 稔 (岡山県)水野真樹 BシャープSTAFFジャンパー (山形県)伊藤克弘 (東京都)小ロ一法 (奈良県)今井健生 (広島県)菅原洋祐 (愛媛県)坂本和秀 Cメガホン (東京都)鈴木雅之 (山梨県)川久保大樹 (長野県)山本武一 (京都府)森 竹紘 (鳥取県)菊地理一朗 DストⅡイベントフラッグ (千葉県)猪狩友則 (大阪府)浦田隆史 Eマーカーセット (山形県)清野一男 (栃木県)大嶋靖浩 (京都府)山地将朗 (大阪府)佐々本博和 松尾泰之 (石川県)三宅 勤 (山口県)白石和大 (愛媛県)加藤和人 5CD「マニキュア団」 (東京都)小林宏昭 塚田浩一

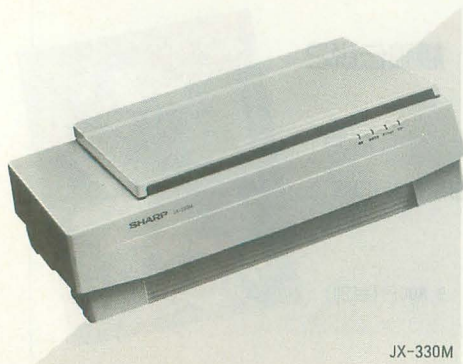
(敬称略)

以上の方々が当選しました。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

カラーイメージスキャナ
JX-330X/M/N/P
シャープ

JX-330M

シャープはカラーイメージスキャナの新機種「JX-330」シリーズを発売する。

基本解像度は600dpi，デジタル画像処理により最高2400dpiまでサポートしている。読み取り時の解像度の指定は30～2400dpiの範囲で0.01dpi単位でできる。読み取り速度は，A4サイズ(300dpi)のモノクロ原稿で約1秒(従来機比約30分の1)，カラー原稿で約13秒(従来機比約6分の1)となっている(同社調べ)。また，一方向光源読み取り方式により，凹凸のある原稿でも鮮明な読み込みが可能。インタフェイスはSCSI-2を標準装備。付属品はSCSIケーブルと対応マシンの読み取りソフト(「JX-330N」のみSCSIインタフェイスボード付属)。

オプションの「JX-3F」シリーズを利用することで，ネガやポジなどの透過原稿の読み込みも可能。

X68000用「JX-330X」は6月25日発売予定。価格はどれも178,000円(税別)。

〈問い合わせ先〉
シャープ㈱　☎043(297)1221

3倍速CD-ROMドライブ
CXA-450
緑電子

CXA-450

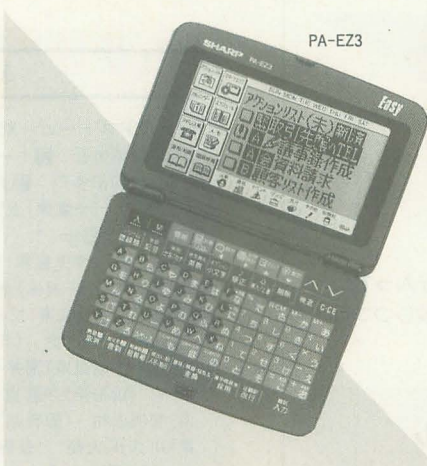
緑電子は3倍速CD-ROMドライブ「CXA-450」を発売した。

本機はデータ転送速度450Kバイト/secの3倍速CD-ROMドライブユニットを採用している。平均シークタイムは約200ms。メモリバッファは256Kバイトを搭載。インタフェイスはSCSI/SCSI-2に対応している。

付属品はI/Fケーブル，AC電源ケーブルなど。デバイスドライバはPC-9801，Macintosh，DOS/V用が用意されている。したがって，X680x0での使用には別途デバイスドライバが必要となる。

価格はPC-9801シリーズ用が79,800円，Macintosh，DOS/V用が89,800円(それぞれ税別)となっている。

〈問い合わせ先〉
緑電子㈱　☎044(989)7633

電子ビジネス手帳
PA-EZ3
シャープ

PA-EZ3

シャープは電子ビジネス手帳「PA-EZ3」を発売した。

本機の主な特長としては，スケジュール管理用に優先順位がひと目でわかるアクションリストの採用や，収支の内容をメモしておけば，集計を行ってくれるマネーチェック機能などがあげられる。ほかにも約40,000語の国語辞典，約44,340語の英和辞典，約16,000語の和英辞典に加え，約350例の英会話文例を搭載している。

また，カレンダーは3種類が用意されており，シークレット機能によりスケジュールの保護が可能になっている。アドレス帳には，住所，氏名などのほかに似顔絵を入れることができる。時計機能は通常の表示以外に，主要120都市のなかから1都市の日付時刻表示が行える。

画面にはFSTN液晶(96×64ドット)が採用され最大16桁×5行の表示ができる。メモリ容量は32Kバイト(ユーザーエリア約28Kバイト)。また，通信機能では4ピン端子で同社の電子手帳とのデータ交換が可能になっている。

大きさは閉じたときの状態で，117mm(幅)×81mm(奥行)×10.9mm(高さ)。

価格は29,800円(税別)。

〈問い合わせ先〉
シャープ㈱　☎043(297)1221

無停電電源装置
US-500L
ティアック

US-500L

ティアックは個人での使用を想定した無停電電源装置「US-500L」を発売した。

本機は停電，電源瞬断，電圧変動などの発生時に，内蔵バッテリーからインバータを通して電力を供給する。ほかには，落雷などによって起こるサージノイズを吸収する回路を内蔵し，接続した機器を保護する機能を搭載している。

出力容量は500VAで，約5分間300Wの電力供給を行う。電源異常発生中(バッテリー電力使用中)にはアラームブザーで警告を発する。また，バッテリーには小型シール鉛蓄電池を使用し，充電時間は12時間以

内(6時間で約80%の充電)である。停電バックアップ出力は2系統用意されている。

大きさは380mm(幅)×335mm(奥行)×45mm(高さ)で，重さが6.5kg。

価格は49,800円(税別)。

〈問い合わせ先〉

ティアック(株)　☎0422(52)5013

フォトジョイプリンタ “Print-it” NC-1 富士写真フイルム

NC-1

富士写真フイルムはフォトジョイプリンタ“Print-it”「NC-1」を発売した。

同機はビデオ出力のあるすべてのAV機器からカラープリントが得られる。明るさや色合いなどの調整は画面に表示されるメニューを見ながら行う。また，マルチ画面分割機能により4画面または16画面の画像を1枚のプリントにできる。入出力端子はビデオとS映像の2つずつ。

記録用紙にはインクカートリッジや現像液などが不要で，熱に反応して自己発色し，フルカラー画像が得られるTA方式(直接感熱記録方式)が利用されている。

プリントペーパーの大きさは100mm×140mmでプリント時間は約70秒/枚。プリントサイズは78mm(縦)×100mm(横)のフルサイズと50mm(縦)×64mm(横)のポストカードサイズの2種類。画素数はそれぞれ，468×704ドット，352×384ドット。

価格は115,000円で専用の感熱記録用紙は20枚入りで1,200円(それぞれ税別)。

〈問い合わせ先〉

富士写真フイルム(株)　☎03(3406)2111

デジタルスチルカメラ VC-1100 オリンパス光学工業

オリンパス光学工業はデジタルスチルカメラ「VC-1100」を発売する。

本機は静止画の撮影を行い，そのデータの送受信が可能。撮影に関しては現在販売中の「VC-1000」とほぼ同様で，フルオート撮影，2倍ズーム，マクロ撮影などの機能を搭載している。撮影した画像はカラー液晶ファインダによって，その場で確認することができる。

VC-1100

静止画の記録はPCカードによって行う。このカードはJEIDA(社)日本電子工業振興協会)規格に対応し，画像圧縮方式はJPEGを採用，ファイル形式はDOSフォーマットとなっている。記録できるコマ数は，2Mバイトのカードの場合，標準圧縮で31コマまで可能。記録モードはノーマル，エコノミー，スタジオ，ファインの4種類。

伝送に関しては，外部モデムを介して一般公衆回線を使って行う。もちろん携帯電話を利用しての伝送も可能。伝送先は「VC-1100」同士やパソコン(専用ソフトが必要)，NIFTY-Serveなど。

大きさは109mm(幅)×155mm(奥行)×62mm(高さ)で，重さは約600g(バッテリーは除く)。

価格は348,000円(税別)。

〈問い合わせ先〉

オリンパス光学工業(株)　☎0426(91)7657

DATデッキ D-05 パイオニア

D-05

パイオニアはDATデッキ「D-05」を発売する。

本機はサンプリング周波数を96kHzにしたHSモードを搭載している。これにより録音再生周波数は44kHzまで可能。そのほかのモードにはSPモードとLPモードがあり，サンプリング周波数はそれぞれ48，32kHz，録音再生周波数は22，14.5kHzまで可能。HSモード以外のノーマルサンプリングでも再生時には「レガート・リンク・コンバージョン」(音楽信号の波形再現性の向上を目的とした技術：記録された信号データ間を，自然界の音などがもつ周波数成分を再現するのに適した関数曲線で結ぶ)を搭載したことで，従来以上の広帯域再生を実現した。各モードの最大録音時間は，120分テープでHSモードが60分，SPモードが120分，LSモードが240分である。

また，曲名，タイトル，アーティスト名などを1曲当たり60文字記録できるキャラクタレコーディング機能を搭載している。テープの最初の部分に記憶するTOC(Table of Contents)情報では，それぞれの曲の開始時間などの情報を記録して，曲のトータル時間，各曲の時間，残量時間が表示できる。さらに，このTOC情報をもとに，高速AIサーチが可能。

価格は89,800円(税別)。

〈問い合わせ先〉

パイオニア(株)　☎03(3491)8181

INFORMATION

ヒューマン コンピュータ エンタテインメント・コンテスト'94 ヒューマンクリエイティブスクール

ヒューマンクリエイティブスクールは「ヒューマン コンピュータ エンタテインメント・コンテスト'94」を実施する。

同コンテストではコンピュータを使用した作品および企画を集め，優勝者を選出する。応募部門は以下の4つ。

1　プログラム部門
2　CG映像部門
3　ミュージック部門
4　企画，アイデア部門

応募条件は未発表のもので自作のもの。何点でも応募は可能。個人またはグループで制作したもの。応募作品の返却は行われない。作品の応募については各種メディア(フロッピーディスク，MO，ビデオテープ，カセットテープなど)で行う。ただし企画部門についてはペーパーに限る。

また，応募作品の著作権については，すべて応募者に帰属する。

募集期間は1994年4月19日～9月30日まで(当日消印有効)。

〈問い合わせ先〉

ヒューマンクリエイティブスクール内

HUCEC運営事務局　☎0422(23)1111

FILES

Oh!X

このインデックスは，タイトル，注記――著者名，誌名，月号，ページで構成されています。初夏の趣さえ感じるこの頃ですが，新しい生活にはもう慣れたでしょうか。たまの休みには仲間とどこかへ行くのもいいかもしれませんね。

参考文献

I/O　工学社
ASAHIパソコン　朝日新聞社
ASCII　アスキー
コンプティーク　角川書店
C Magazine　ソフトバンク
電撃王　主婦の友社
PIXEL　図形処理情報センター
マイコンBASIC Magazine　電波新聞社
My Computer Magazine　電波新聞社
LOGIN　アスキー

一般

▶ついに出た　パワーマッキントッシュ
「Power Macintosh」の詳しい内容を紹介。――斉藤勉，ASAHIパソコン，4.15号，14-17pp.

▶ハッカーに，なりたい
悪く語られることも多いハッカー。その真の姿を探るべく綴られたインタビュー集。――福山栄子・笠原利香，ASAHIパソコン，4・15号，86-95pp.

▶機械用言博物館　7
動詞を軸にパソコンの使い方を解説する連載。「変換する」をテーマに日本語入力を扱う。――荻窪圭，ASAHIパソコン，4・155号，96-97pp.

▶デイトナUSA
セガのAM2研が技術の粋を注ぎ込んだ「DAITONA USA」の魅力を徹底解剖。コース攻略アドバイスつき。――編集部，電撃王，5月号，32-39pp.

▶ゲーム白書1993
電撃王編集部が独自に集計したデータをもとに，家庭用ゲーム機の1993年の売り上げの傾向と結果を分析する。――編集部，電撃王，5月号，87-102pp.

▶Hardware
ソニーのコンピュータ用MDドライブ，パソコンで3D画像を実現するキットのニュースなど。――編集部，コンプティーク，5月号，94-95pp.

▶CD-ROMドライブ＆ソフト
マルチメディアに欠かせないCD-ROM。その魅力を紹介する特集。現行機種一覧表つき。――編集部，マイコンBASIC Magazine，5月号，8-9，33-41pp.

▶Let's Try MULTIMEDIA!
マルチメディア関連の周辺機器をレポートするコーナー。第1回目として，「PhotoCD」とビデオプリンタを使いフルカラー環境作りに挑戦する。――編集部，マイコンBASIC Magazine，5月号，45-47pp.

▶'94AOUショー・レポート
千葉の幕張メッセにて行われた「AOU1994アミューズメントエキスポ」の模様をレポート。――編集部，マイコンBASIC Magazine，5月号，148-151pp.

▶Arcade Game Graffiti
アーケードゲームの歴史を編纂する。1979年にポストインベーダーを狙ったゲームたちを紹介する。――編集部，マイコンBASIC Magazine，5月号，152-155pp.

▶特集　忘れ去られたファンタジー
「伝統的」ファンタジー世界とはどんなものかをレクチャーする。――編集部，LOGIN，8号，99-111pp.

▶架想楽園へ行こうVer.2.02
バーチャルリアリティの進化の鍵を握る日本の研究者を訪ね，日本のVR研究の現状をインタビューする。東京大学の広瀬研究室を訪ねる。――中田宏之，LOGIN，8号，162-165pp.

▶知らなかったほうがよかった世界
アメリカのメーカー，アタリを取り上げる。アタリを愛する人なら持っていたいアイテムも紹介。――編集部，LOGIN，8号，166-169pp.

▶特集I　パワーマックの秘密に迫る
「Power Macintosh」の秘めたるパワーを紹介。――編集部，ASAHIパソコン，5・1号，14-25pp.

▶機械用言博物館　8
日本語入力シリーズのラストを飾る。この回のテーマは「鍛える」。――荻窪圭，ASAHIパソコン，5・1号，104-105pp.

▶日本ソフトウェア大賞'93
3月25日に決定した日本ソフトウェア大賞'93の「トルネコの大冒険」など各賞を発表。――編集部，LOGIN，9・10合併号，12-15pp.

▶THE CHOICE!
LOGIN編集者が選んだ，1993年のゲームソフトの各賞を発表。――編集部，LOGIN，9・10合併号，16-19pp.

▶特集　突撃メディアミックス大作戦！
パソコン主導のメディアミックスを，関連メーカーへのインタビューなどを交えて考える。――編集部，LOGIN，9・10合併号，139-155pp.

▶THE NEWS FILE
「Power Macintosh」や「ビデオCD for X680x0」発売のニュースなど。――編集部，LOGIN，9・10合併号，156-163pp.

▶L.I.A.
CDやビデオがついている新しい形のテーブルトークの登場など，海外からのニュースを掲載。――編集部，LOGIN，9・10合併号，164-165pp.

▶知って得するハイビジョンのひみつ
知っているようで知らないテレビ受像機の原理やハイビジョンの実態にせまる。――編集部，LOGIN，9・10合併号，190-193pp.

▶架想楽園へ行こうVer.2.02
日本のバーチャルリアリティ都市，岐阜の実態を紹介する。なぜ岐阜か，またその実力はなど。――中田宏之，LOGIN，9・10合併号，202-205pp.

▶特集　パーソナルDTP'94
DTPを家庭へ普及させる鍵などを議論し，DTPやそれを支える技術を紹介する。――古橋優ほか，I/O，5月号，43-60pp.

▶特集　パソコン高速化作戦！
パソコンの速度を決める要因の解説や，高速化で実際になにをすればよいかを考える。――さるさ ほか，I/O，5月号，72-82pp.

▶カタログの読み方
パソコンや周辺機器のカタログを題材に，語句の解説と，カタログのへそまがりな読み方，質問点の捜し方など。――田嶋孝之，I/O，5月号，83-91pp.

▶マルチメディアの行方　5
「MACWORLD Expo/Tokyo'94」の様子や，3DO発売後のソフト開発はどうなっているのかなど。――奥野雅之，I/O，5月号，129-132pp.

▶UNIXアベニュー　3
UNIXの紹介。この回のテーマは「標準入出力」で，住所録を作りながらファイルの作成・更新の方法を解説する。――大森俊太郎，I/O，5月号，138-141pp.

▶新製品情報解読マニュアル
カタログなどに使われている用語などについて解説している。――田島孝之，I/O，5月号，別冊付録

▶特集　しあわせの大容量記憶装置
さまざまな記憶装置の現況や今後どうなっていくのか，HDDの賢い使い方などを紹介する。最新HDD一覧表つき。――編集部，ASCII，5月号，273-292pp.

▶INTERCOOLED　第1回
次世代ゲーム機や家庭用マルチメディア機に関する情報を紹介する連載。この回は，3DO用のゲーム「Total Eclipse」などのレポート。――編集部，ASCII，5月号，340-343pp.

▶魅惑のニューテクノロジー　第2回
最新技術を解説するコーナー。今回は動画再生に関する技術。――編集部，ASCII，5月号，348-353pp.

▶COMPUTER PERSPECTIVE ＜後編＞
「A COMPUTER PERSPECTIVE――計算機創造の軌跡――」から，抄訳を抜粋。1920年代後半から1940年代までのコンピュータ開発の歴史をたどる。――編集部，ASCII，5月号，375-382pp.

▶バカパパのモノを買い物
音声認識をするリモコンや外出先でカギをかけたかどうか確認できるユニットなど，優れたグッズを紹介する。――バカパパ，ASCII，5月号，410-412pp.

▶アスキーロードテスト No.54
ザウルスの試用レポートの2回目。アプリケーションを使う。――宮野友彦，ASCII，5月号，468-469pp.

▶特集　CD-ROMドライブ装置の選択と活用
CD-ROMドライブ選択のポイントを考える。内蔵，外付け主要機種の紹介あり。――編集部，My Computer Magazine，5月号，14-28pp.

▶未来派パソコン通信　最終回
SMDの後継機種として最大通信速度28.8KbpsのV-Fastモデムが発売される。それについてのモニターレポート。――原田洋平，My Computer Magazine，5月号，144p.

▶ビジネスマンのための情報管理術
ザウルスを使う人のための活用法。前回に続き，覚えていると便利な機能を紹介する。――塚田洋一，My Computer Magazine，5月号，151-154pp.

X1/turbo/Z

X1シリーズ

▶ロボット艦隊宇宙船シミュレーション・WAR
ロボット艦隊を使ってUFO艦隊を迎撃するシミュレーションゲーム。──アー君，マイコンBASIC Magazine，5月号，110-112pp.

X68000

▶X68新聞
電波新聞社の「アルゴスの戦士」，SX-WINDOW用ワープロソフト「EGWord SX-68K」などを紹介。──編集部，LOGIN，8号，132-133pp.

▶GAME BUSTERS!
パソコン用ゲームの攻略法を徹底解説。X68000用は，「ドラゴンナイト4」が登場。──編集部，LOGIN，8号，196-199pp.

▶電撃王全ゲームインデックス
各機種の発売情報を収録。X68000用は「スーパーリアル麻雀PIV」などが登場。──編集部，電撃王，5月号，6-8pp.

▶今月の電撃王
注目作，話題作を紹介する。X68000用「アルゴスの戦士」など。──編集部，電撃王，5月号，9-13pp.

▶新作王
新作情報を紹介。X68000用は「麻雀航海記」「大魔界村」などが登場。──編集部，電撃王，5月号，161，163pp.

▶電撃新作予定表
ゲームなどの新作データ。X68000用は「レッスルエンジェルス3」「餓狼伝説SPECIAL」など。──編集部，電撃王，5月号，182p.

▶SUPER SOFT EXPRESS
各マシンの最新ソフト情報。X68000用は「大魔界村」「スーパーリアル麻雀PIV」「あすか120% BURNING Fest.」「アルゴスの戦士」。──編集部，コンプティーク，5月号，33-34，36，45pp.

▶SQUARE PANELS
パイプドリーム風のアクションパズル。──高橋潤，マイコンBASIC Magazine，5月号，113-115pp.

▶COURAGE HEART
タイムトライアルのドライブゲーム。──高橋秀之，マイコンBASIC Magazine，5月号，116-118pp.

▶ロマンシング　サ・ガ2～七英雄バトル～
ミュージックプログラム。NAGDRV2＋GS音源用。──上古仁志，マイコンBASIC Magazine，5月号，127-129pp.

▶SUPER SOFT HOT INFORMATION
新作紹介ページ。X68000のコーナーでは，「大魔界村」「アルゴスの戦士」など。──編集部，マイコンBASIC Magazine，5月号，とじ込み付録12p.

▶NEW SOFT
「大魔界村」「ドラゴンナイト4」など，各機種用の新作ゲームを紹介する。──編集部，LOGIN，9・10合併号，20-33pp.

▶未確認クリエイターズ
プログラム部門にX68000用の「カワード」と「フォーナイツ」が入選した。──編集部，LOGIN，9・10合併号，185-186pp.

▶AV STRASSE
SX-WINDOW用開発キット「Workroom SX-68K」や，SX-WINDOW用ワープロ「EGWord SX-68K」の紹介。──編集部，ASCII，5月号，361-364pp.

▶ONLINE SOFTWARE INDEX
大手ネットにアップロードされたソフトを紹介。Human68kのドライブをコントロールする「drvctrl.x」など。──編集部，ASCII，5月号，455p.

▶SX-WINDOW対応 EGWord
シャープから発売された「EGWord SX-68K」を詳しく紹介する。──高橋雄一，My Computer Magazine，5月号，44-45pp.

▶なんでもQ&A
シャープから発売されたSX-WINDOW用ソフト「Workroom SX-68K」「EGWord SX-68K」などの特徴を答える。──シャープ，My Computer Magazine，5月号，162-163pp.

▶HOBBY EXPRESS
X68000用ゲームソフト「ドラゴンバスター」「エキサイティング・アワー/出世大相撲」「ジオグラフシール」など，各機種のゲームの説明。──稲垣宗彦ほか，My Computer Magazine，5月号，179-193pp.

▶簡単で本格的な2次元画像作画教室　4
「MATIER」を使った作画教室の第4回。今回はラフスケッチの描き方について解説する。──長谷川一光，PIXEL，5月号，72，154-158pp.

▶SX-WINDOWプログラミング　第7回
いままでに作ったプログラムをもとに，複数のウィンドウと簡単なメニューを表示する。──吉野智興，C Magazine，5月号，132-139pp.

ポケコン

PC-E500

▶あんたが大将ー勇者になるためにー
パズルゲーム。壁を押して道を造りながら迷路を抜ける。──市之渡剛，マイコンBASIC Magazine，5月号，121p.

新刊書案内

知の技法
小林康夫/船曳建夫編
東京大学出版会刊
☎03(3811)8814
A5判　283ページ
1,500円（税別）

いくら新刊とはいえ，大学のサブテキストとして作られたもの，しかも大学の出版会が出版したテキストが一般の書店で平積みになる機会というのはそう多くはないうえ，それが手に取ってみたいものである可能性はまずないものだ。だが，「知の技法」はちょっと違った。「東京大学教養学部「基礎演習」テキスト」だからややこしい専門書とはひと味違う。教養学部の「基礎演習」として，さまざまなジャンルの講師が学問のさわりだけを大学1年生向けにうまく記述しているからわかりやすいし，そこに描かれる「頭と身体の使い方」の具体性や多様性は大学生以外にも十分読むに値するからだ。古い大学人には，いまの学生にはこんなことから教えねばならないのか，と嘆く人もいるだろうが，おかげで大学生でなくても面白く読める本ができたのだから，これは感謝せねばなるまい。そもそも，ここに書かれている「将来どのような専門領域を研究するにしても，かならず身につけておかなければならないきわめて基本的な知の技法」がきちんと教えられることはまず（私の経験した限りでは）なく，学生が自ら発見せねばならないことだった。「文系の大学生」に向けて書かれたものであり，最初の章では「文系」という言葉が頻出するが，文系理系を問わず，学問として研究しているしていないにかかわらず，「かならず身につけておかなければならないきわめて基本的な知の技法」に大きな違いはない。特に刺激的なのは「学問の行為論」「認識の技術」「表現の技術」と3部構成になっているうちの，第2部。さまざまな執筆者が自分の専門領域に関して例を挙げて具体的に，しかもわかりやすく語っているところだ。それが，マドンナの写真集の解釈であったり，統計の読み方であったり，漱石のテキストの解釈だったり，難民報道の落とし穴だったり，フィールドワークだったりする。興味あるジャンルだけに目を通しても面白いこと受けあいだ。（K）

裸のアインシュタイン
ロジャー・ハイフィールド/ポール・カーター著
古賀弥生訳
徳間書店刊
☎03(3433)6231
四六判　414ページ
1,800円（税込）

本書はアインシュタインの伝記である。もちろん皆さんもその名前をご存じであろう。相対性理論の創始者であり，20世紀を代表する科学者である。しかし，彼の実生活については神秘のベールに包まれていた。本書はスキャンダラスでこれまでの彼のイメージを壊すものである。皆さんのなかには，科学の世界で偉大な業績を残しているのだから，実生活など問題ないと思うかもしれない。だが，本当にそうであろうか。彼もあくまで人間なのだ。彼がすべてのことを相対化しようとしたように，今度は我々が彼自身を相対化する番ではないだろうか。

非日常実用講座
ジャンボ・ジェット機の飛ばし方
非日常研究会著
同文書院刊
☎03(3812)7777
A5変形判　204ページ
1,300円（税込）

本書は非日常的な状況下におかれたときに，それに対処するためのマニュアル本である。状況設定は，面白可笑しく書いてあるが，これは読みやすくするためのご愛敬であろう。登場する乗り物は，F1からヨット，戦闘機までと，陸海空さまざまである。また，イラストも多く，欄外の注釈も気がきいていて，臨場感にあふれている。

日常の刺激に飽き足りないとか，世紀末の状況に不安を覚えるというあなたにお勧めする。ただ，実際にそれぞれの乗り物を動かす可能性はほとんどないため，本当に正しいのかどうか確かめられないのが残念である。だからこそ非日常なのだが。

QUESTION and ANSWER

自作ゲームなどで使用したいのですが，X-BASIC上からマウスカーソルの形状を変えるにはどのようにすればよいのでしょうか。

長崎県　高山　忠信

X-BASICにはマウスカーソルの形状を変更する命令はありません。ここは定石どおりに外部関数でやってみようかということになります。

では外部関数を使うとして，マウスカーソルのパターンを変更する方法ですが，これはIOCSコールに用意されていますので処理自体は非常に簡単です。ということで，マウスカーソルのパターンを変更する外部関数を作成して問題解決にあたることにしました。

まずマウスカーソルのパターンを定義するIOCSコールはMS_PATSTです。これからX-BASICでのコマンド名称はmspatに決定しました。続いてマウスカーソルの定義に必要となるパラメータを決めます。

マウスカーソルのパターン（16×16ドット×2）は配列に入れて渡すことにして，あとは，このパターン内のどこの座標をマウス座標での原点（0，0）とするか，さらにマウスカーソルは20個定義できますので，カーソル番号0～19のどこに定義するかを指定できるようにします。

さらに，マウスカーソルパターンの複数定義にともなって，カーソル番号0～19のどれを画面表示に使うか指定するmsselコマンドも同時に作成することにしました。

以上の点を踏まえて作成した外部関数のプログラムがリスト1です。

as ms.s

リスト1

```
  1: *
  2: * マウスカーソル変更外部関数
  3: *
  4: * mspat(x;char,y;char,cur_pat;ary1_char,ms_number;char)
  5: *
  6: * mssel(ms_number;char)
  7: *
  8:         .xdef       _mspat
  9:         .xdef       _mssel
 10:
 11:         .include    iocscall.mac
 12:         .include    fdef.h
 13:
 14: * infomation table
 15:
 16:         .dc.l   x_init
 17:         .dc.l   x_run
 18:         .dc.l   x_end
 19:         .dc.l   x_sys
 20:         .dc.l   x_brk
 21:         .dc.l   x_ctrl_d
 22:         .dc.l   x_res1
 23:         .dc.l   x_res2
 24:         .dc.l   ptr_token
 25:         .dc.l   ptr_param
 26:         .dc.l   ptr_exec
 27:         .dc.l   0,0,0,0,0
 28:
 29: x_init:
 30: x_run:
 31: x_end:
 32: x_sys:
 33: x_brk:
 34: x_ctrl_d:
 35: x_res1:
 36: x_res2:
 37:         rts
 38:
 39: ptr_token:
 40:         dc.b    'mspat',0
 41:         dc.b    'mssel',0
 42:         dc.b    0
 43:
 44:         .even
 45:
 46: ptr_param:
 47:         dc.l    mspat_par
 48:         dc.l    mssel_par
 49: mspat_par:
 50:         dc.w    char_val    *   char_val = $0004
 51:         dc.w    char_val    *   char_val = $0004
 52:         dc.w    ary1_c      *   ary2_c   = $0034
 53:         dc.w    char_val    *   char_val = $0004
 54:         dc.w    void_ret    *   void     = $ffff
 55: mssel_par:
 56:         dc.w    char_val    *   char_val = $0004
 57:         dc.w    void_ret    *   void     = $ffff
 58: ptr_exec:
 59:         .dc.l   mspat
 60:         .dc.l   mssel
 61:
 62:         .text
 63:         .even
 64:
 65: _mspat:
 66:         move.l  4(sp),d0        * offset X
 67:         move.l  8(sp),d2        * offset Y
 68:         move.l  12(sp),a0
 69:         move.l  16(sp),d1
 70:         bsr     mspat_entry
 71:         rts
 72: mspat:
 73:         moveq.l #0,d0
 74:         moveq.l #0,d2
 75:         move.b  15(sp),d0
 76:         move.b  25(sp),d2
 77:         move.l  32(sp),a1
 78:         tst.b   d0              * offset X はマイナスでないか
 79:         bmi     offset_err
 80:         tst.b   d2              * offset Y はマイナスでないか
 81:         bmi     offset_err
 82:         cmpi.b  #16,d0          * offset X は16以上でないか
 83:         bcc     offset_err
 84:         cmpi.b  #16,d2          * offset Y は16以上でないか
 85:         bcc     offset_err
 86:         moveq.l #0,d1
 87:         move.b  45(sp),d1
 88:         tst.b   d1              * マウス番号はマイナスでないか
 89:         bmi     msnum_err
 90:         cmpi.b  #20,d1          * マウス番号20以上でないか
 91:         bcc     msnum_err
 92:         cmpi.w  #64,8(a1)
 93:         bne     err_ret
 94:         lea.l   10(a1),a0
 95: mspat_entry:
 96:         lea.l   ms_x,a1
 97:         move.w  d0,(a1)         * マウス座標X
 98:         move.w  d2,2(a1)        * マウス座標Y
 99:         lea.l   pat_area,a2
100:         moveq.l #15,d0
101: loop:
102:         move.l  (a0)+,(a2)+
103:         dbra    d0,loop
104:         IOCS    _MS_PATST
105:         moveq.l #0,d0
106:         rts
107:
108: _mssel:
109:         move.l  4(sp),d1
110:         bra     mssel_entry
111: mssel:
112:         moveq.l #0,d1
113:         move.b  15(sp),d1
114:         tst.b   d1              * マウス番号はマイナスでないか
115:         bmi     msnum_err
116:         cmpi.b  #20,d1          * マウス番号20以上でないか
117:         bcc     msnum_err
118: mssel_entry:
119:         IOCS    _MS_SEL
120:         moveq.l #0,d0
121:         rts
122: *
123: * エラー終了
124: *
125: offset_err:
126:         lea.l   offset_mes,a1
127:         rts
128: msnum_err:
129:         lea.l   msnum_mes,a1
130:         rts
131: err_ret:
132:         lea.l   err_mes,a1
133:         rts
134:
135:         .data
136:
137: ms_x:
138:         ds.w    1
139: ms_y:
140:         ds.w    1
141: pat_area:
142:         ds.b    64
143:
144: offset_mes:
145:         .dc.b   '座標は0～15を指定してください',0
146: msnum_mes:
147:         .dc.b   'マウス番号は0～19を指定してください',0
148: err_mes:
149:         .dc.b   '配列の大きさが64バイトではありません',0
150:
151:         .end
```

lk ms.o /oms.fnc

としてms.fncを作成します（インクルードファイルがカレントディレクトリになければ，as.xの/iスイッチでインクルードファイルのあるディレクトリを指定します)。あとはms.fncをBASIC.Xを置いているディレクトリにコピーしておき，BASIC.CNFに，

FUNC = MS.FNC

の1行を加えます。

これでX-BASIC上からマウスカーソルのパターンを変更する関数が使えるようになります。表1に関数リファレンスをまとめておいたので参考にしてください。一応，リスト2にサンプルプログラムを紹介しておきますので，表1とあわせてご覧ください。

なおX-BASICとCでは，引数の受け渡し方法が違うだけですので，リスト1ではCのライブラリにも対応してみました。今回作成した外部関数を含んでいるX-BASICのプログラムをコンパイルするときは，リンク時にms.o（as ms.sで作成されるもの）を一緒にリンクするようにしてください。

（朝倉 祐二）

表1

MSPAT
書式　mspat(x,y,cpat,num)
引数　char(x,y,num),char型1次元配列名(cpat)
戻り値　void
機能　マウスカーソルのパターンをcpatで定義したパターンに変更します。
　x,yには16×16ドットのパターンのうち，どの座標をマウス座標での原点とするかをx,yともに0〜15の範囲で指定します。
　char型1次元配列cpatにはマウスカーソルの16×16ドットのパターンデータを影にするパターン，表示するパターンの順番で各32バイトずつ合計64バイトを定義します。
　numには定義するカーソル番号を0〜19の範囲で指定します。
注：詳しくはプログラマーズマニュアルの_MS_PATSTの説明を読んでください

MSSEL
書式　mssel(num)
引数　char(num)
戻り値　void
機能　表示するマウスカーソル番号を0〜19の範囲で指定します。
注：詳しくはプログラマーズマニュアルの_MS_SELの説明を読んでください

リスト2

```
 10 /*
 20 /* マウスカーソル変更サンプル
 30 /*
 40 int x,y
 50 width 96:msarea(0,0,767,511)
 60 dim char c(64)={&HFC,&H3F,&HFC,&H3F,&HFC,&H3F,&HFC,&H3F,
 70                &HFC,&H3F,&HFC,&H3F,&HFC,&H3F,&HFC,&H3F,
 80                &HFC,&H3F,&HFC,&H3F,&HFC,&H3F,&HE0,&H7,
 90                &HF0,&HF,&HF8,&H1F,&HFC,&H3F,&HFE,&H7F,
100                &H0,&H0,&H3,&H80,&H3,&H80,&H3,&H80,
110                &H3,&H80,&H3,&H80,&H3,&H80,&H3,&H80,
120                &H3,&H80,&H3,&H80,&H3,&H80,&H1F,&HF0,
130                &HF,&HE0,&H7,&HC0,&H3,&H80,&H1,&H0}
140 mspat(7,15,c,0) /* マウスパターンの(7,15)はマウス座標の(0,0)
150                 /* マウス番号0にパターンを設定
160 mssel(0)        /* マウス番号は0を選択
170 mouse(1)        /* マウス表示
180 mspos(x,y)      /* マウス座標取得
190 locate 0,0:print "x:";x,"y:";y
200 goto 180        /* ブレークが押されるまでループ
```

付録ディスクに入っていたSX-BASICについてよくわからないところがあるので質問します。

Text1.forecolor = 1

はテキストアイテムに関するプロパティ設定で，

Text1.move = 10,10,100,100

はメソッド，とのことですが，このプロパティとメソッドの違いがよくわかりません。どちらもアイテムのなんらかの値を設定するものだと思うのですが。

東京都　小嶋　高志

基本的にプロパティ設定，メソッドともにアイテムの持つ属性（文字の描画色，アイテムの座標）を設定するものです。一般に，メソッドにより属性を変更した場合，アイテムがなんらかの動作を行います(生成，消去，移動)。これに対し，プロパティではあまり大きな動作が起こらないことになっています。

で，位置が変わるのと色が変わるのでは，なにが違うのかといわれても困るのですが，

●forecolor　描画色を表すデータの値を変更すると，たまたま画面上で変化が起こった（あくまでも，後者は付随事項)。

●move　アイテムを移動させるために，たまたま引数が必要だった（メインは移動)。

のように考えてください。以上のような理由で見た目は似ていますが，目的は若干異なるのです。

メソッドは，アイテムを現在とは違う状態へと導くことに主眼が置かれていますが，プロパティ設定は，あくまでもプロパティを変更するためのものです。ですから，メソッドによりプロパティが変化することはありません。一部の例外を除いて，メソッドの結果を調べるプロパティもありません。

プロパティ設定の結果は，記録に残ります。プロパティ参照（print Text1.forecolorで現在の描画色を得ることができます）を用いて，プロパティがどのように設定されているかを調べることはできますが，メソッド参照という機能はありません。

……と，私は考えていますが，SX-BASICではあまり厳密な区分をしていません。アイテムの座標を表すプロパティ（現バージョンではサポートされていません）は，moveメソッドを用いても変更されますし，このようなメソッドが今後表れないという保証はありません。

ですから，いまのところ両者を厳密に区別する必要はありません。正直にいうと，オブジェクト指向な人のために設けた便宜的な区別にすぎません（もっと正直にいうとある処理系からパクってきたものです)。プロパティ参照はできても，メソッド参照はできない，ということを覚えておいてくだされればそれで結構です。（石上　達也）

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を挙げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに解答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので電話番号も明記してください。

宛先：〒103　東京都中央区日本橋浜町
3-42-3
ソフトバンク株式会社出版部
Oh!X編集部「Oh!X質問箱」係

FROM READERS TO THE EDITOR

春の暖かさから，夏の暑さへと変わる今日この頃。そろそろ新しい環境にも慣れ，いろんなことに挑戦し始めた頃でしょうか。うっとうしい梅雨に負けないために，しっかり食べて体力をつけておきましょうね。ただ太りすぎには……。

◆4月号から始まったローテク工作実験室，なかなかよいです。私のようなハードには興味はあるが，よくわかっていないような人間にはちょうどいいです。　中村　吉邦(22)神奈川県

初心者には難しいところもありますが，なかなか好評です。これからに期待ですね。

◆3月号にタコ足配線の話が載っていたので，それについてひと言。コンセントには極性があります(アースとそうでないほうの2つ)。タコ足配線をするときは極性を統一するとよいです。具体的にいうと，ひとつのタップに差すコンセントのコードを見るとコードの片側に白い線が入っていることがあるので，その向きをすべて同じ向きにします。こうすると，AV製品のノイズは大幅に低減できます。

青木　泰一郎(19)東京都

さっそく家のタコ足配線を調べてしまいました。

◆ひえ～っ！　X68000ACEのメイン基板上のMESSIANというICが火を吹いた。すごかった。ところで特集が「X68000の仲間たち」から「SX-WINDOWの活用」になったのはなぜ？　実は楽しみにしていたのです。　田川　和義(20)岐阜県

今月号でちゃんと特集してますから許してくださいね。

◆変わるべきなのは学生か教育方法論か？　僕は前者が70%，後者が30%変わらないといけないと思う。教わるのではなく学ぶのである。

長瀬　孝司(18)岐阜県

もっともな意見だとは思いつつも耳が痛い。学生の頃を思い出してしまいます。

◆「エキサイティングアワー/出世大相撲」のレビューのタイトルはエグい。「裸の男同士……」って，そりゃそうなんですけどね。でも，前にも横内氏が「悪魔城ドラキュラ」のレビューのとき「ムチとロウソクと俺」なんてタイトルをつけていたし，瀧氏はローテク特集のとき「下着は黒が好き」なんていってたし……。ひょっとしてOh!X編集部は「ソドムの館」なんだろうか？　すごくイヤ……。池田　譲太(25)大阪府

私もイヤ……ってことで，そんなことはありません，ご心配なく。

◆昨年，某店で「エトワールプリンセス」を新品特価6,800円で買った。昨日，同じ店で「エトワールプリンセス」を中古特価9,800円で売っていた。こんなことがあっていいのだろうか？

久米　豊信(25)大阪府

市場の原理とはいえ，なんか納得できないのが……。

◆SX-WINDOWのシャーペン.Xで卒論を書いたが，RAM 4 Mバイトではすぐメモリ不足になり，周りのMacintoshで書いている連中に比べて2倍以上の時間がかかってしまった。図のペーストができず，すべて手作業になったのが敗因だった。「Easydraw」を買っておくべきだったと思った。　大谷　英克(24)東京都

そしてあなたの手元には「Easydraw」が……。その前にメモリですか？

◆C言語でプログラムをしてもなにをやっているのか気になって，アセンブラのソースを吐き出させ，結局アセンブラでコーディングしてしまう私はいったい……。最初に覚えたのがZ80だったのが悪いのだろうか。いまだにBASICすら理解できません。　水丸　淳(20)埼玉県

BASICが理解できなくてもアセンブラが理解できればぜんぜん問題ないような気がします。それとも自慢ですか？

◆友人が「卵に穴を空ければ電子レンジでゆで卵を作ることができる」といっていたので5mm程度の穴を空け，マグカップに卵まるごといれて2分間加熱。取り出して，できたかなと思い覗いてみると卵が爆発，顔面に直撃。左目の中には黄身が入ってしまいました。もし，殻が入っていたらと思うと……。不幸中の幸いだったと思います。友人に「だましたな！」と詰めよると，殻を割って，マグカップに入れて黄身に穴を空ければ大丈夫という意味でいったのだそうです。　河合　英敏(21)北海道

それでできたものはゆで卵というのでしょうか。なんか違うような……。

◆最近，イヤなことも笑ってすませられるようになりました。また一歩仙人に近づけた，そんな気持ちです。こまんたれぶー。

永井　邦彦(24)愛知県

いまこの瞬間にも，また一歩仙人へ近づいていることでしょう。がんばってください。

◆6月号が出る頃，長男が生まれる予定です。2月まで「たぶん女の子です」と医者にいわれていたのですが，エコーにはっきりと○○が写って「間違いなく男の子です」といわれたそうです。名前はすでに廷哲(ただあき)と決めています。長女の智恵ちゃんともどもよろしくお願いします。P.S.いままで3回載せていただいたものは，妻がコピーしてアルバムに貼っています。　石川　仲宰(35)福岡県

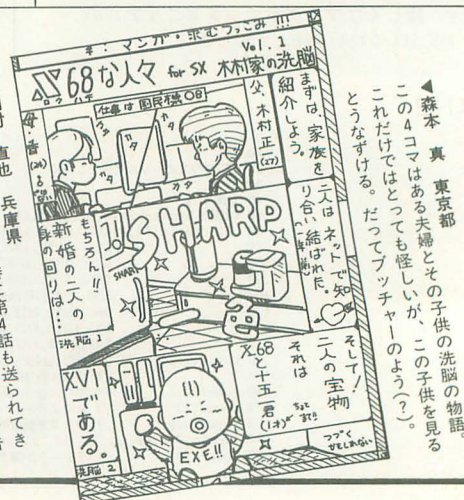
子供たちの成長とともにOh!Xをよろしく。

◆眉毛が薄くて困っている人へ。僕は眉毛がつながってしまうためときどき剃っていますが，最近すぐ生えて困ります。ただ，もともと薄い人だと毛穴がふさがって生えなくなるかもしれないので，剃らずに抜くとか，先のほうを切るとかするといいんじゃないですか。

吉成　雷(18)千葉県

◆4月号の清野さんへ，僕も同じことで悩んでいましたが，啓芳堂製薬の「ミクロゲン・パスタ」という塗り薬を先月から使用し始めて，現在かなり生えてきました。一度お試しください(写真

◀岡村　直也　兵庫県
これが第3話ですが，同時に第4話も送られてきました。来月号に掲載の予定です。これって予告(笑)。ちなみに学が主役のバージョンです。

◀森本　真　東京都
この4コマはある夫婦とその子供の洗脳の物語。これだけではとっても怪しいが，この子供を見るとうなずける。だってブッチャーのよう(?)。

がないのでわからないかもしれませんが)。なんか通販のコーナーみたいだ。

滝水　正樹(18)福岡県

地道な努力をしてみるか，昔からある薬に頼るか。あとはあなたの自由です。

◆卒論やっと終わったし，さあ，試合に向けてがんばるぞう。毎日約20kmくらいのランニングに筋トレやって，組み手の練習やって……。きついけどやりたいことをやれるのはとてもいい気分です。5月21，22日が勝負だ！

松嶋　竜(23)東京都

この勝負がなにかは4月号を見てくださいね。それにしても毎日20kmのランニングですか。私だと家に帰りつけないかもしれません。

◆やっぱり靴を洗うには亀の子たわしでしょう(笑)。あ……ちくちく……うふふ。

藍原　和久(23)東京都

どこか別の世界の人なのでしょうか。

◆某番組のプレゼントでゲームソフトが当たった。が，なぜか3.5インチ版が送られてきてしまった。悔しかったので再び応募したら，同じソフトを当てることができた。まさか1カ月の間に2度も当選するとは思ってもみなかったけど，また3.5インチ版が送られてきたら……。

増田　和通(20)東京都

これは，3.5インチFDDを買え……という某番組の陰謀なのでは。

◆4月号で余ったチョコボールを編集部に送れば食べてくれるといっていましたが，その後どうですか？　まさか編集部全員デブになってしまったとか……。　神保　公一(18)東京都

残念ながらチョコボールは送られてきませんでした。したがって，デブにはならずにすんでいます。

◆コンピュータ部の部長は僕らしい。それはそれでよいのだが，今度入ってくる後輩にX68000ユーザーがいるかどうか，そっちのほうが問題だ。誰もいないと文化祭のとき，学校にX68000 XVIをもっていくことになってしまう。誰か入ってこないかなぁ。　澤田　恭幸(16)千葉県

きっとがんばって勧誘したことでしょう。成果はいかがでしたか。

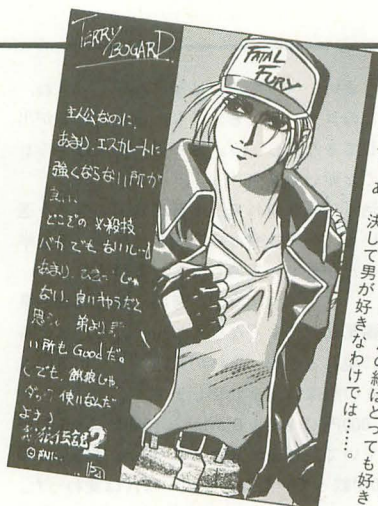
◆最近，パソコン通信をよくやっていて，とうとう悪名高き「6ケタ」をやってしまいました。どうしよう……バイトのお金がみ～んななくなってしまいます。皆さんはどのくらいお使いですか？　星野　こずえ(19)千葉県

そんな……私の家賃よりも高いんですか。私なら生きていけないかもしれません。

◆とある店でアルバイトをしたら，5日目店長に「君はこの仕事に向いてない」といわれた。ショック！　佐怒賀　英一(26)神奈川県

大丈夫。ほかに向いてる仕事がありますよ……たぶん。

◆休日出勤の仕事中に書いている。去年の仕事がやっと終わった。職場には私を含めて2人しかいない。すごく寂しい。

宮越　良幸(21)神奈川県

◀藤澤　篤　奈良県
このテリーってなんか目が涼しげですね。4月号のザンギエフといい藤澤さんの絵はとっても好きです。あ，決して男が好きなわけでは……。

◀松居　啓樹　富山県
3月号の橋本さんのチュンリーを見て，送ってくださったようです。なぜかこの絵を見て中国っぽいと思ってしまったのは私だけですか。

ご苦労様です。これからやっと今年のお仕事なんですね。ということは今年の仕事が終わるのは，来年のいまごろ……ごめんなさい。

◆友人との会話。「『創価学会』って草加市にあると思わなかった？」「思ってた，思ってた」「レーザープリンタはレーザーで紙を焼いて印字するとか……」「え，それは正しいんじゃ……」お願いだからもう少し勉強してくださいな。彼はレーザープリンタは感熱プリンタの一種だと思っていたそうだ。　中村　健(24)埼玉県

でも自分では当然だと思っていたことが，勘違いだったなんてことはよくありますから，正しいことを教えてあげてくださいね。

◆いまさらながらうちのX68000PRO改にSASIハードディスク40Mバイトが載った。金欠の私にはこんなハードディスクでも6,000円なら涙が出るほど嬉しいのでした。でも，さすがに狭かったりする。残り数Mバイト……(涙)。

河上　博仁(17)埼玉県

次は200Mバイトのハードディスクでしょうか。その前にバイトかな。

◆私の部屋にはサボテンとマリモとハエトリソウがいます。プログラミングで疲れたときに水をやったり，音楽を聴かせてあげたりします。先日，アセンブラで「ジュリア集合描画プログラム」ができました。これも柴田氏とサボテンたちのおかげです。　小林　健一(20)群馬県

それぞれに名前はついていないんですか。ついていたら，今度教えてくださいね。

◆「パンプキン・オ・レ」という飲み物を見つけたので飲んでみた。味はまあまあだった……と思ったのだが，友人2名には評判がよくなかった。私の味覚が変なのだろうか。見かけたらぜひ試してみてください。西尾　昌人(20)愛知県

そうえいば「ソーダス(チョコバナナ風味)」なんてのを見かけましたが……。

◆これを書いている数日後には就職のためX68000を抱えて新潟から福島へ引っ越しです。死んでもキミ(X68000)とは一緒だ。離さないゾ。　吉田　晴彦(18)新潟県

するとこの本はきっと福島の空の下で読まれているんですね。

◆行き先が決まりました。これからは新しい町へ出てひとり暮らしで勉強です。親に多大な負担をかけてしまうことになりますが，自分で決めたところなので無駄にならないようにがんばりたいです。　天達　雄一(18)京都府

新しい場所へ移ったらまた連絡をくださいね。

◆友人M子さんの通うK女子大の落とし物コーナーに高級腕時計や毛皮などが並んでいたと聞き，“さすがはお嬢大学”と驚いていたら，翌日うちの大学の落とし物コーナーにセーラームーンのキーホルダーと下敷きが並んでいました。うちの大学っていったい……。このことは決して彼女にいうまいと心に誓うのでした。

大畑　佳史(20)兵庫県

世の中には本当にいろんな人がいますから。でも毛皮などはともかく，そのキーホルダーや下敷きは受け取りにいけないような気が……。

◆山で採ってきたあけびの種を植えてみようかなと思っております。別に庭作りや盆栽が趣味というわけではないのですが，庭を荒れたまま放っておくのも寂しいですし。それになにか植えておけば，以後少しは手入れをするようになるだろうという実に甘い考えもあって……。まあ，数年，十数年後を考えて(んな大げさな)，いまは雑草採りからコツコツやっていこうと考えています。　渡辺　久孝(27)大阪府

たまには庭いじりなんかをすると心落ち着くかもしれませんね。しかし，そんな庭なんて……。うらやましい

◆某大阪のFM局に出演しました。で，DJと話をしているときに「悪魔城ドラキュラ」の“やめてっ，いやぁ～”っていうPCMを鳴らしてやった。この放送，誰か聴いていません？

牛島　真一(18)大阪府

なにも知らずにそんな放送を聴いた人はどう思ったんでしょうか。ちょっと気になります。

◆春になり，新団員が入ってきた。まだ仮入団だが全員入ってくれると嬉しい。ただ，私のパート(トロンボーン)にひとりもこないのが寂しい。このままでは，一応キャリアだけはあるが

下手な自分と最近コンバートした素人の２人だけになってしまう。困った。
安尾　文教(26)愛知県

いま頃は新しいメンバーと楽しくやっているんでしょうか。

◆名古屋に来て４日目，やっぱり鳥取とは違うな〜。
田村　孝文(18)愛知県

どんなところが違うのか今度教えてくださいね。

◆あの「アランケイ」という本を読んだ。そこにはコンピュータの未来が書かれてあった。「ダイナブック」への道はまだまだ遠い。
要田　計治(19)広島県

皆さんが夢見るコンピュータの未来はどんなものですか。

◆３月号を買い忘れてしまいました。だから，本屋へ注文したのですが，入荷は２〜３週間後。先輩には３月号を自慢され，買えなかったことをバカにされ耐えに耐えた３週間。やっと手元にきて「ばんざーい」と思ったのも束の間。３月号ってアンケートがついてたんですね。また「ちゃだワ」に参加できなくなってしまいました。ぐすん。
森　美和子(24)大阪府

来年の参加を心よりお待ちしております。

◆自分で考えていることが翌月の質問箱に載っていたりするので非常に役に立ちます。また，そのときわからなくても，あとで読み返すと役に立つのでOh!Xは捨てられません。
田口　実(20)神奈川県

理解してしまったらその号は捨ててしまうのでしょうか。そんなことせずに大切にしてくださいね。

◆社会人になって１年がたったが，最近ではほとんどX68000を起動しなくなってしまった。“社会人は忙しい”というのはやはり本当なのだとつくづく実感しています。
川道　武志(23)大阪府

忙しいなかでいかに時間を見つけていくか，たまには触ってあげてくださいね。

◆イカクッキーというものを食べました。クッキーがイカ味ではなく，クッキーにイカが入っているのです。味は……。
小高　幸夫(21)埼玉県

うまい……ってことはないんでしょうね，このコメントだと。でもいったいどこが出してるんでしょう。知っている人はよろしくお願いします。

◆生活リズムが崩れて，１日25時間の生活を送っている。一般社会と同期が合わない。中途半端に１日が長いってのもちょっとねえ。
音羽　進(19)宮城県

編集部内にもそんな人が……。今日は何時に現れるやら。これが冗談かどうかは皆さんの想像におまかせします。

◆20,000円のWave Blasterに10MHzの68000が載っていて，ビデオ入力ユニットには25MHzの68020が載っているなんて……時代は変わった。
吉田　征二(20)宮城県

本当に時の流れは早いもの。ふう。

◆某T◯WNSで「サムライスピリッツ」の発売が決まった。魔法さん，じらさないで早くX68000にも移植してください。もちろんジャパンホーム◯デオなんかに負けないデキでお願いしますよ！
田中　康治(22)東京都

てっきり５月あたりには発表されるかと思いましたよね。だって某誌４月号のソフトハウスインフォメーションのコーナーで「◯◯◯◯◯◯◯◯◯」速報？　なんてありましたから……。

◆何故にスーパーリアル麻雀PIVの原画集がソフトバンクから……。硬派な会社だと思っていたんですが違ったのでしょうか？　でも買ってしまった。
一場　英輔(18)東京都

ずっと硬派でいると疲れるものです。たまには軟派なものも気休めに必要ってことで皆さんも見てみては。

◆最近，ぬいぐるみ取りが熱い。300円でけっこう取れたりする。Ｔシャツを絞って糸で巻いてあるヘビなどは，なかなかイケる。話は変わるが，我が県では４月から原付自転車の運転免許が18歳以上でないと取れなくなるらしい。確かに，高校生の事故の割合は多いと思うが，このことで泣いている人も多いのではないだろうか。
長谷川　祐之(17)新潟県

その時点で免許を取っていた人はどうなるんでしょうか。まさか没収……。

◆３月25日は大学の卒業式がありました。それと同時に「ぷよぷよ」の発売日でもありました。卒業式は気もそぞろで出て，謝恩会の前にちょっと抜け出して買ってきました。もちろん夜はX68000とスーパーファミコンでぷよぷよ大会でした。
村上　淳一(22)福岡県

どうせならスーパーファミコンではなくメガドライブのほうが美しい(？)。

◆ロサンゼルスに約１カ月ほどホームステイをしてきました。アメリカは２回目ですが行くたびにアメリカのよさが見えてきます。そういえば成田でエンジンが取れた飛行機は僕たちが乗ったやつとか……。ちょっとびっくりしました。あと，サンフランシスコに行ったとき，外人さん(男)に「ハァ〜イ，ユーア キュート」なんていわれて，身の危険を感じ思いっきり走って逃げました。いま思うといい経験だったかなと思っています。なにはともあれ無事に帰ってこれてよかったです。
井上　潤一(20)福井県

そうか，アメリカのよさっていうのは男性に「かわいい」って声をかけてもらえるところですか。そんなわけは……。

◆ついてない。値下がりを待っていた170Mバイトのハードディスク。待ちきれずに買ったら次の週に2,3000円から19,800円に値下がりした。全部「餓狼伝説２」のディスクの入れ替えのせいだ。
沢田　剛規(24)東京都

八つ当たりですね。でも気持ちは痛いほどわかります。

◆「ゲームウォーカー」という「東京ウォーカー」の別冊を見ました。そのなかのX68030の紹介文に「かつてはNECとシェアを分けあっていたシャープの製品」「初心者向けというよりは，いずれはそのパソコンを使いこなそうと思っているマニア向けの機種」ということが書いてありました。なかなかわかっている人が書いているなと思いつつも，現実を見つめさせる結果となりました。「シャープのパソコン」の灯を絶やしてはならないと強く感じました。
阿部　文明(21)東京都

それにしても「ゲームウォーカー」の表紙は「東京ウォーカー」とそっくりでした。コンビニで見かけてもすぐには区別がつきません。

◆「やっとMOを買ったぞ。これでハードディスクのバックアップが……メディア買い忘れた」「ふう，ハードディスクのバックアップが終わった。これでいつでも引っ越しできる」(会社の人)「研修期間中は着替えと布団以外はもってこないでください」嘘だといってほしかった。
宮本　俊彦(23)愛知県

研修期間だけならなんとか我慢ができますよね。もしも期間が６カ月だったりしたら……。

◆田村健人さんへ。私は昔，買ったばかりの自転車でバナナの皮を踏んで転び，自転車を壊したことがあります。
山中　政宣(20)三重県

バナナの皮を踏んで転んだ人っていたんですね。初めて聞きました。

◀岩瀬　貴代美　福岡県
就職してからなにかと大変なようですががんばってくださいね。イラストの洋服は会社の制服でしょうか。で，右手にあるのは鞄ですよね。

◀占部　哲彦　広島県
ゲームセンターでは，かなり痛めつけられているみたいですね。でも，X68000版が発売され，それで３姉妹に恨みを晴らしている頃でしょうか。

◆忘れないうちに書いておこう。1月号のイラストのリー・ラプラダは有名なボディビルダーのことです。念のためいっておくが私はアドンな人ではない。それからコナミの「リーサルエンフォーサーズ」の「うっちゃだめ」は「Put your hands up!」(手を上げろ)と聞き取れました。　沢地　文彦(23)兵庫県

ライターの某氏はショーン・レーのほうが好きだといっていました。でも，誰がいったい知っているんだろう。

◆三半規管の弱い僕は「ジオグラフシール」をやるとすぐに気持ち悪くなる。面白いゲームなだけにこれは悲しい。ちなみにスーパーファミコンの「マリオカート」でも気持ち悪くなる。やっぱり悲しい。　河野　太郎(21)東京都

すると体感ゲームなんかもずいぶんつらいのでしょうね。

◆医師国家試験を無事終了した。難しくなるのではとの前評判だったが，フタを開けてみると昨年と同じ程度でむしろやさしいぐらいだった。6割で合格というのは受験生からするとありがたいが実際に働きだすと4割ちかくが誤診するということである。これからも勉強し続けようと思う。　尹　忠秀(24)長崎県

なにかの記事で読んだのですが，権威のある医師が退職したあと自分の診察の記録を発表したことがあったそうです。その記録によれば誤診率は2割。我々が聞くととても高そうな誤診率ですが，医師に聞くとやはり低いそうです。それ以下になるようにがんばってくださいね。

◆それにしても車の免許を最近取らなくてよかったと思います。「ドラクエ」みたいに並びたくはないですからね。　藤原　彰人(23)岡山県

本当に大変らしいですね。

◆パソコンショップでバイトをしています。それで最近思うのですが，世の中は本当に不景気なのでしょうか。今回の米騒動と同じで，ただの買い控えをしているだけではないのでしょうか。というのは，小・中学生が万札を出して買い物をしているんです。普通，不景気なら小遣いに影響が出そうなものなのに……。世の中が変わってしまったのでしょうか。子供は不景気だと思っているのかなぁ。　平田　崇(19)愛知県

就職状況を見ていると，やっぱり不景気なのかなと感じることはありますけど，実際のところはどうなんでしょうね。

◆うちの大学は1年生から2年生になるときに大きな関門があります。入学時には学科未定のまま入るので，進級のときにその学科を決めるのです。もちろん成績でです。よって，いま現在2年生にはなれそうですが，学科はまだ未定……ああ。　橋本　直己(20)東京都

成績によって決まるというのは，成績のいい順番に好きな学科が選べるということでしょうか？　それならまだしも，上位から順番に学校側が学科を決めていったら……。そんなことはないですよね。

◆俺は40時間以上寝ないでいると，とってもハイになる……らしい。　村上　元章(19)静岡県

らしい，ということは自分にはすでに記憶がないわけですね。ひょっとしたら夢遊病かもしれませんね。

◀酒井　強　三重県
ちゃだワには間に合いませんでしたので，常連の方のなかには心配しておられる方もいたみたいです。どうやらお子さんはまだみたいですね。

◆受験が終わり，X68000やOh!Xを引っ張りだした。Oh!Xにはカビが生えていたが，フロッピーには生えていなかった。　細田　千晶(18)埼玉県

なんとかわいそうなOh!X。カビが生えるような環境に1年間も放って置かれたんですね。しかし，フロッピーに負けてしまうとはまだまだ修練が足りないようですね。

◆「ストリートファイターIIダッシュ」はSELECT(MODE)＋6ボタンで“求む！　対戦プレイ”が出せる。　相原　泰三(21)鹿児島県

知らない人はお試しください。

◆パチンコで6日連続のプラス。ハードディスクが買えるかも　大上　幸宏(21)鹿児島県

200Mバイトなら，いまでも買えるかも。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

売ります

★東京システムリサーチのメモリボード「Xsimm10」に4MバイトSIMM2枚を載せたもの(8Mバイトのメモリボード)を42,000円程度で売ります。また，MIDIボード「CZ-6BM1A」を8,000円程度で売ります。どちらも箱，説明書，付属品すべて揃っています。連絡は往復ハガキでお願いします。〒161　東京都新宿区西落合2-7-20　檜佐剛欣(17)

★東京システムリサーチのメモリボード「Xsimm10」と30ピン1MバイトSIMM(70ns)2枚と30ピン4MバイトSIMM(70ns)2枚，以上すべてを50,000円(送料込み)で売ります。連絡は往復ハガキでお願いします。〒131　東京都墨田区八広1-8-4　折田　貴弘(21)

★ローランドの音源モジュール「CM-32L」とシステムサコムのMIDIボード「SX-68MII」と電波新聞社のオーディオミキシングケーブルをセットで45,000円前後で売ります。それぞれ単品で売ることも可です。どの製品も新品同様で箱，マニュアルなどの付属品もすべてあります。連絡は希望価格を書いて往復ハガキでお願いいたします。〒487　愛知県春日井市高森台7-1-1-308-208　山本　克(19)

買います

★カラーイメージユニット「CZ-6VT1」を送料込み35,000円で買います。説明書と付属品があれば箱はなくてもかまいません。連絡は往復ハガキでお願いします。〒156　東京都世田谷区上北沢5-7-12　伊藤　恵介(24)

★カラーイメージユニット「CZ-6VT1-BK(黒)」を25,000円で買います。説明書と付属品があればかまいません。連絡は官製ハガキでお願いします。〒157　東京都世田谷区北烏山4-31-10　桜井　暢(39)

★アイテックのSASIハードディスク「TX-80」を20,000円くらいで買います。連絡は往復ハガキでお願いいたします。〒520-25　滋賀県蒲生郡竜王町美松台1-4-1　井上　健(21)

バックナンバー

★Oh!Xの1990年4,6〜11月号を送料込みで各2,500円にて買います。切り抜きがないものをお願いします。連絡は官製ハガキでお願いします。〒731-01　広島県広島市安佐南区大町東3-7-34　高橋　一俊(27)

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今月は4月号の内容に関するレポートです。

●アプリケーション紹介＋SX-BASICの解説，という感じだったけど，少々物足りない。「Easydraw」の記事はわりと普通の使い方を書いてあるだけだし，ウィンドウデザイナの説明は新しい情報があまりない。「SX-WINDOW v3.0 テキストマネージャ解析結果」のようなありがたい記事がもっとあるといいんだけど……。

あと，中野氏の「人間不親切起源論」の「美しくないものに慣れるのは悪いことだ」の一文，これはよかった。確かに最近のSX-WINDOWは操作性が美しくなくなっているようだ。安易にメニューを多用している雰囲気があるし。ウィンドウデザイナも気をつけて（少々美しくない）。

石田　伯仁(20)　X68030, MZ-731, PC-8801mkIIMR, PC-E200　神奈川県

●別々に進行していたことが偶然タイミングが合ってしまい，じゃあ特集にしましょうか，というようなノリだったのではないでしょうか。「EGword」もでるし「Double Bookin'」も出るし……。特集のそれぞれの記事を繋ぐことはただひとつ「SX-WINDOWでやっています」だけだったのではないでしょうか。バラエティのない雑誌なんてあまり楽しくないですが，てんでんバラバラというのもどうかと思います。「特集」としてまとめるならもう少し題材を絞ってやってほしいと思うのです。それぞれの記事が押さえるところは押さえていただけに，余計バラバラだという気がしたのかもしれませんが。

でも考えてみればウィンドウシステムなんてのはただの「環境」なわけだし，そのなかでなにをやるか，となるとかなり広い範囲のことになってしまうわけですね。だから，今回の特集は，例えるなら「X68000の活用と題して，ゲームのレビューやX-BASICの活用，それにWP.Xの使い方の記事など」という感じだったと思います。すると，SX-WINDOWも「ひとつのアプリケーション」という感じから抜けてきた，てなことでしょうか。次からはSX-WINDOWをメインとした特集ではなく，「SX-WINDOWのなかの云々について」という形でないとひとつの特集に収まりきらなくなりつつあるのではと感じました。だとしたら頼もしい限りです。

林　大助(18)　X68000 SUPER, PC-8801mkIIFR　神奈川県

●「1993年度GAME OF THE YEAR」ですが，今回はコナミのがんばりで独壇場でした。最近はソフトの数も減少し，これはと思えるソフトが少なくなっているなか，ハードを使いこなしたゲーム作りに誰もが納得できたのではないでしょうか。これからもいいゲームを作り続けてほしいものです。

八亀　圭一(19)　X68000 PRO　神奈川県

●初めてCGAコンテストの会場に行ってきた。思ってた以上にパワフルな作品が次から次へと出てきた。にもかかわらず，グランプリ該当作品は「なし」だという。やや疑問に思いながら家に帰り，さっそくビデオを見てみる。確かに飛び抜けた作品はなかったように思えた。しかし，全体のレベルは確実にUPしている。それは審査する側にもいえることだろう。実によい傾向だと思いつつ，初心者には厳しい状況だな，とも思う(泣)。でも負けないぞ！

吉岡　洋明(21)　X68000 PRO II, PC-8801MA, FM-NEW7　埼玉県

●「ローテク工作実験室」は今後どういったものを作るのかはわかりませんが興味があります。手先が不器用だが，なにかを作ってみたいということでプログラミングに興味をもってパソコンを始め，最近ではハードウェアを作ることにも興味がわいてきました。そこにこの連載ですから，就職活動が終わったらチャレンジしてみたいと思います。失敗するとダメージが大きそうですが，趣味でやるのはそれくらいでないと面白くないのかもしれませんね。

北風　保(22)　X68000 ACE　東京都

●「ローテク工作実験室」いいですね。MIDI音源は高い！　とあきらめていたら，こんな方法があったのですね。目からうろこ（コンタクトレンズともいう）！

適当なスイッチング電源さえあれば，こんな簡単な結線でMIDI音源，しかもSC-55相当品が手に入るとあれば，お買い得度は確かに10ですね。出来合いの基板に周辺部品をくっつけるだけでできてしまうところは，まさにローテクの極致。WBの入手先も（さりげなく）書いてあるし，WBのコネクタのピンアサインも説明してあるし，これだけわかればすぐにでも（WBを買うお金が工面できれば）作れそうです。

第1回から嬉しい内容で，次回も期待しています。

野原　賢次(33)　X68000 ACE-HD　埼玉県

●「石の言葉，言葉の夢」についてである。コードレスは非常に便利だ。テレビやリモコンにコードがついていたら，うっとおしい（昔はついていた）。電話がコードレスになることは，以前には予想もできなかった。ファミコンのコントローラやヘッドフォン，アイロンなどもコードレスのものが多くなってきたが，私がいちばんコードレスにしてほしいのは電源だ。電源コンセントに電気製品をいちいち接続するのは面倒くさい。電源がコードレスになると快適な生活を送ることができるのだが。

松永　孝治(24)　X1turbo model30, MZ-80C, PC-9801N, AMIGA1200　鳥取県

●ビデオ入力ユニット「CZ-6VS1」ですが，Macintosh用に開発されたようでどうも面白くありません。ま，時代の趨勢がそうなので仕方がないでしょう。言葉ばかりが先行した「マルチメディア」ですが，パーソナルユースでビデオデータが扱えたりすると，雰囲気出てきますよね。

橋本　和典(27)　X68000 XVI, Macintosh LC-520　東京都

ごめんなさいのコーナー

5月号　こいのぼりPRO-68K

P.47　MOD.X ver.1.06

付録ディスクに収録されたソースのインクルードファイル名に間違いがありました。

_slashlib.h → slashlib.h

以上のように修正してコンパイルしてください。ご迷惑をおかけしたことをお詫びいたします。

5月号　(善)のゲームミュージックでバビンチョ

P.96　ビクターエンタテインメントの名前が間違っていました。関係者の方々にご迷惑をおかけしたことをお詫びいたします。

バグに関するお問い合わせは
☎03(5642)8182(直通)
月～金曜日16：00～18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

友達たくさんできるといいな

▶どうもお待たせいたしました。4月号で特集の予定でしたが，今月号に延びての登場となりました「X68000と仲間たち」。

最近(といっても数カ月くらい前からですが)，X68000の周辺機器関係がずいぶん騒がしくなってきています。以前紹介したメモリーボード「Xsimm10」やビデオ入力ユニット「CZ-6VS1」,「発売予定のカラーイメージスキャナ「JX-330X」やX68030用アクセラレータ「040turbo」などさまざまですね。

周辺機器というのは自分たちのパソコン環境を快適なものにするため必要なものですよね。パソコンは，それだけならただの箱です。そのパソコンを使って自分たちがしたいこと(ゲーム，CG，音楽，プログラミングなど)をやろうとするときに，本体に付属のものだけではなにかと不便なものです。だから，ジョイスティックやタブレット，MIDI音源，コプロセッサなどが必要になるのです。最近では，ゲームをするにもメモリが必要ですし，ハードディスクもないと辛くなってきていますしね。4月号のアンケートの回答に外部記憶装置が多いのもそのせいかもしれません。

お世辞にも，X68000の周辺機器は充実しているとはいえません。接続などに問題があるものも事実あったりします。でも，自分のやりたいことを快適にするために試行錯誤してみませんか。

▶特集で，ジャストのアクセラレータも取り上げる予定でしたが，開発が遅れているようで，今回は見送りという結果になってしまいました。楽しみにしていた方には残念でしたが，発売されたおりには，機会を見つけてレポートする予定です。

▶「X68000マシン語プログラミング」「石の言葉，言葉の夢」は著者多忙のため，「DōGA CGアニメーション講座」はある計画の準備のためにお休みです。また，「“実戦！”ゲーム作りのKNOW HOW」は著者がBGマッピング処理(応用版)を鋭意制作中ですので，いましばらくお待ちください。

投稿応募要領

●原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。

●プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。

●ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。

●投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒103 東京都中央区日本橋浜町3-42-3

ソフトバンク出版部

Oh!X「㋢㋮㋴」係

SHIFT・BREAK

▶道を歩いていたら，南アメリカの装束をしたバンドが演奏していた。CDを売っていたので買ってみた。予想外に当たり。嬉しかった。インディオの人権擁護運動家を支援するためにドイツで録音されたものらしい。悲しげな旋律を聴いていると，なぜか郷愁をそそられる。やっぱりモンゴロイド同士，音楽のセンスがどこか通じてるのかな。(E.K.)

▶今月は死にそうだった。24時間以上も働いてから寝て，また24時間以上働くという生活をずいぶんした。毎月これだったら死んでしまうだろうなあ，きっと。そういえば，純正モニタでも垂直同期を68Hzにできることを発見。インタレスで1024×700超が出来ることがわかった。やり方は，いまは文字が足りないから，また今度ね。(瀧)

▶某日池袋にて。「あの……高橋さん……ですよね？」「はい，そうですけど？」「やっぱりぃ！　いつもOh!X読んでますぅ」「あ，そりゃどうも(なんで僕の顔知ってるんだろ？)」「握手してもらえます？」「べつにかまいませんけど」ここで彼女の目つきが変わった。「あなたの健康と幸せを3分間お祈りさせていただきます」「あ゛……!?」(哲)

▶なんと，(で)氏のおかげでMOを使える環境になった。多少お金ができたので，憧れのオシロスコープを買いに行く。アクセラレータは，なんとなく完成間近（しかし，オシロスコープなしで作ってたというのもひどい話だ)。寮にオシロスコープとユニバーサルプログラマを持ち込んだら，みんな怪しむだろうなぁ。(石)

▶これを書いてるいまはGW直前！　実は(で)もこれから旅に出ます。行き先は常夏の島，プーケット。今回はなにもかも忘れてぱーっと過ごすのだ！でもってしこたまメシ食って，ゲームセンターでゲームして……なんかいつもと変わらんような気がしてきた(苦笑)。あ，ショートプロは休まないから安心してね。んでは，行ってきま～すっ！(で)

▶結局JAMP(Just Another Multimedia Player)の先陣を切った3DO REAL。熱い期待をしていただけに失望も大きい。ご自慢の3Dグラフィック機能が生かされていないと思っていたが，それは誤解だった。CD-ROMからの垂れ流し2Dアニメーションも，空間2次元時間1次元だから立派に3Dなのだ！　……というのがジョークで済むことを切に願う。(A.T.)

▶サラリーマン時代の同僚が結婚したというのでパーティへ出かけた。会場に着くと3年ぶりの同僚に「現れるだけで受けるのはお前だけだ」といわれる。が，会話がうまくかみ合わない。よく見ると，彼らは私が退職したときと変わってない。サラリーマンの空間と時間の関係は私のとは違うらしい。その関係はリーマン方程式で得られるのだ(ウソ)。(K)

▶日差しがとても気持ちいい。なぜか自分が活動しているときの外は暗い。最近，河川敷にもなかなか行けない。今年もMTBのシーズンが始まったというのに……。よし，GWにはのんびり(？)スポーツ三昧といこう。そして休みの最後は平塚に出撃だ(またもや編集長よりチケットを譲りうけた)！

(編集長に感謝のサンフレッチェファンの高)

▶連休前だというのにパスポートが切れている。貨幣交換しようにも，忙しさのあまり世間さまと顔を合わせられない。予約も手続きもすべて人任せ。出発5日前なのに宿泊場所もスケジュールさえも知らない。こんなんで旅立てるのか？　計画たてたり，わくわくしてその日を待つのも旅行の楽しみのひとつなのに。まあいいや，入稿終わったもんね。(ふ)

▶引っ越しました。引っ越し先はアパートだけど，いままで住んでいたワンルームマンションの倍以上の広さがあるので，とてもうれしい。家に帰るとごろごろ遊んで広さを満喫しています。家賃はかかるけど，通勤が楽になったからいいや。でも「デイトナUSA」が近くのゲームセンターにないのは残念だなあ。(杉並区民のJ)

▶新しいパソコンを買った。とりあえず玩具だから得体の知れないメーカーのものでも気にしない。速けりゃいいというもんじゃないが，高解像のウィンドウ環境がまともに使えるくらいは必要だ。ま，CPUはそのうち改善できるもんだ。へなへなのキータッチは安物だからしかたないか……。とりあえず強化電源だな。(すでに半分バラして使っているU)

▶最近はB5サイズのサブノート(香港製の怪しいヤツ)を持ち歩いている。初代DynaBook用のバッグに入れるとスペースがたっぷり空く。もっとも，本やら資料やらを詰め込んでしまい，かえって重くなる始末だ。それでも終電間際にはこれを振り回しながら走ることに。ハードディスクのことなんか気にしなくなってしまったけど，大丈夫なの？(T)

microOdyssey

「いずれ，パッケージソフトを作りたい」

これは，僕がパソコンを使い始めた中学生の頃に抱いていた夢である。その当時は，エニックス主催のパソコンゲームコンテストなどが行われ，パソコンドリームという言葉があちこちで目についた時期でもあった。たぶん，そういった言葉に触発されたんだろうが，好きなゲームを作って大金が手に入るなら，これはとってもおいしいことではないか？　それに，コンテストに入選している人たちは，ごく普通のパソコン少年だし，ひょっとしたら僕にもチャンスが……うん，できる可能性があるならやってみよう。やり方もなにも知らないただひとりの少年が，パソコンがそこにあるということだけで夢を見ていた。

こうして，毎日，夢を見ながらパソコンに向かっているうちに，パソコンを使うことに楽しさを見つけ，熱心にプログラムを打ち込んだり，友達と1日中X1用「ゼビウス」に興じる生活を送るようになる。まあ，熱心にやっていたといっても，好きなことだけを中心に，しかも知識のつまみ食いといった状況だったので，学習進度は遅かった。ようやくゲームというものを完成したのは，高校生のとき。

ただし，その頃はゲームは面白ければいいや，ということだけを考え，ゲームを面白くするためには，どんな要素がゲームに必要かまで深く追求することはなかった。必然的に作るものは，中途半端で内容が薄く，考えただけで行き詰まり投げ出してしまうことも多くなってきた。

さらに，ゲーム自体もどんどん高度に，複雑になってきて，個人の手にはおえなくなってきたと勝手に思い込んだため，いつしか夢は埋もれてしまった。

そして，Oh!Xと深く関わるようになり，プロの制作者ということを意識しはじめてから転機が訪れる。いままでアマチュアだからということで逃げてきた甘えから，そろそろ脱却しなくてはならない，という危機感もあった。別にゲーム制作を本業にするということではないが，ただの趣味で終わらしたくはないのだ。

で，そのプロとしての意識をもって作り始めたのが「SION4」である。

現在，システムとしては「SIONII」と同じものがようやくできてきた（ポリゴン化した「SIONII」は余裕で可能，とホラを吹いておこう）。残る問題は3Dゲームとしてのデザイン。もちろん，技術的に未熟な部分があるため，システム自体に問題がないわけではない。このあたりは力でねじ伏せることも可能なので，穴があってもフォローができる。つまり，解決する目処が立っているということだ。

しかし，「SION4」の世界をどのようにしてプレイヤーに見せるか，ということに関しては，まだ全体像が見えていない。根本的な部分がおろそかになっているのは情けないが，もっともっとアイデアを捻り出さなくては。

いまかいまかと登場を待ちわびている読者の方には本当に申しわけない。完成するまでは，せっせと仕事の合間をぬって，いままでの経験と時間（ついでにOh!Xスタッフも）を使い，プログラミングに明け暮れるだろう。　（J）

1994年7月号6月18日(土)発売

特集　コンピュータミュージック

・SC-88試用レポート
・Mu-1 GS/MUSIC SX-68K製品紹介
・そのほか音楽関連記事

江口響子の実用講座：Photo CD
ビジネスショウ'94レポート
DōGA CGAシステム入門キット
特別付録　5"2HDディスク

予定価格850円

バックナンバー常備店

地域	地区	店名	電話
東京	神保町	三省堂神田本店5F	03(3233)3312
	〃	書泉ブックマートBI	03(3294)0011
	〃	書泉グランデ5F	03(3295)0011
	秋葉原	T-ZONE 7Fブックゾーン	03(3257)2660
	八重洲	八重洲ブックセンター3F	03(3281)1811
	新宿	紀伊国屋書店本店	03(3354)0131
	高田馬場	未来堂書店	03(3209)0656
	渋谷	大盛堂書店	03(3463)0511
	池袋	旭屋書店池袋店	03(3986)0311
	八王子	くまざわ書店八王子本店	0426(25)1201
神奈川	厚木	有隣堂厚木店	0462(23)4111
	平塚	文教堂四の宮店	0463(54)2880
千葉	柏	新星堂カルチェ5	0471(64)8551
	船橋	リブロ船橋店	0474(25)0111
	〃	芳林堂書店津田沼店	0474(78)3737
	千葉	多田屋千葉セントラルプラザ店	043(224)1333
埼玉	川越	黒田書店	0492(25)3138
	川口	岩渕書店	0482(52)2190
茨城	水戸	川又書店駅前店	0292(31)0102
大阪	北区	旭屋書店本店	06(313)1191
	都島区	駸々堂京橋店	06(353)2413
京都	中京区	オーム社書店	075(221)0280
愛知	名古屋	三省堂名古屋店	052(562)0077
	〃	パソコンΣ上前津店	052(251)8334
	刈谷	三洋堂書店刈谷店	0566(24)1134
長野	飯田	平安堂飯田店	0265(24)4545
北海道	室蘭	室蘭工業大学生協	0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある『新規』『継続』のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

基本的に，定期購読に関することは販売局で一括して行っています。住所変更など問題が生じた場合は，Oh!X編集部ではなくソフトバンク販売局へお問い合わせください。

海外送付ご希望の方へ

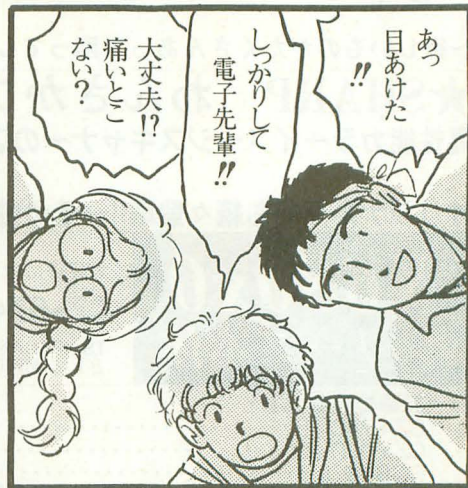
本誌の海外発送代理店，日本IPS（株）にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(3238)0700

Oh!X 6月号
■1994年6月1日発行　定価600円(本体583円)
■発行人　橋本五郎
■編集人　稲葉俊夫
■発売元　ソフトバンク株式会社
■出版事業部　〒103 東京都中央区日本橋浜町3-42-3
Oh!X編集部 ☎03(5642)8122
販売局 ☎03(5642)8100 FAX 03(5641)3424
広告局 ☎03(5642)8111
■印刷　凸版印刷株式会社
©1994 SOFTBANK CORP. 雑誌02179-6 本誌からの無断転載を禁じます。
落丁・乱丁の場合はお取り替えいたします。

満開の電子ちゃん

作・え 岡村祭

第76号以降は5枚組一、五〇〇円に価格改定されます。第72号（94年5月号）は既刊（4/18発売）です。

REDZONE新価格13万円＆コンパクトXVI 9万5千円で登場!!
（持ち込み改造サービスは終了いたしました）

講読方法：定期購読もしくはソフトベンダーTAKERUでお買い求めいただけます。

★定期購読の場合＝購読料第72号（94年5月号）より6ヶ月分7,000円（送料サービス、消費税込）を、現金書留または郵便振替で下記の宛先へお送り下さい。

現金書留の場合：〒171 東京都豊島区長崎1-28-23 Muse西池袋2F ㈱満開製作所
郵便振替の場合：東京 5－362847 ㈱満開製作所

- ご注文の際は、郵便番号・住所・氏名・電話番号を忘れずに記入して下さい。
- 3.5インチディスク版をご希望の方は、「3.5インチ版」とご指定下さい。
- 新規購読の方は「新規」と明記して下さい。なお、特に購読開始号のご指定がない場合は既刊の最新号からお送りいたします。
- 製品の性格上返品には応じられませんが、お申し出があれば定期購読を解約し残金をお返しします。

★TAKERUでお求めの場合＝1部につき1,200円（消費税込）です。

- 定期購読版と内容が一部異なる場合があります。御了承下さい。
- お問い合わせ先　TEL(03)3554－9282（月～金　午前11時～午後6時）

（なお、定期購読版のバックナンバーについては定期購読の方のみご注文を承ります）

中元昌文（兵庫県）

『電倶楽を購読し始めてから…』
オイラ：届いた日の夜に、奇声。「うぉーっ。わっはははー。…おおっ、こ、これは…。だーっはっはっは。…うききー！（後略）」
妹：「おにいちゃんが変だー」
母：「顔が満開」とあきれる。
父：「満開製作所はやっぱり怪しいんとちゃうか」
しかも妹・母・父は『電倶楽』を、語呂がいいから、と『満開』と呼ぶ（いくら言ってもきかない）。
そして、大学の寮にいる兄だけは事態をほとんど知らない。
…すべて実話。
…だからいいんですよ。え？

安いのに親切 **TSUKUMO**

“夏·ツクモ·ザ·バーゲン(夏のボーナス先取り!!)”

～欲しいものがたくさんあって困ってしまう・・・だから、ボーナスで!!～

ツクモラジオCM、ON AIR中!!
東京 ニッポン放送「伊集院光のoh!デカナイト」内
(月)～(木) PM10:30ごろ
名古屋 東海ラジオ「大内ひろのしんの今夜もやっぱりFUNKYパジャマ」内
(月)～(金) PM11:00ごろ
札幌 FMノースウェーブ「SUPER CHOICE」
(月)～(金) PM7:30ごろ
ツクモイメージガール越智静香さんが茶目っ気たっぷりにパソコンをおねだり。
月末には特価品をズバリ！ツクモ特価でお知らせします!!
～これを聴いたら、ツクモに来ないではいられません～

★SHARP わんさかフェアー開催決定!!6/18～30まで★

高性能カラーイメージスキャナーの店頭デモンストレーション。
人気の情報携帯ツール ZAURUS の新製品も登場!!
オリジナル商品も続々登場!!～その他、特価品もいろいろご用意して皆様のご来店をお待ちしております!!～

X680x0 シリーズ本体

大好評に付き、特別セール延長！！なんと 68%OFF です。

CZ-674C-H(X68000 CompactXVI) **超特価¥93,800**

TS-XFDCAを使えば、縦置き5インチモデルX68000ｼﾘｰｽﾞを外部ﾄﾞﾗｲﾌﾞとして使用可能！是非、2台目のﾏｼﾝとしてどうぞ！

お勧めのセット
CZ-674C-H……………………………¥298,000
CZ-607D-BK…………………………¥ 99,800
RGBｹｰﾌﾞﾙ……………………………サービス
ツクモ特価¥148,000

X68030のお勧めの組み合わせ!!
CZ-500C-B…………………¥398,000
240MBハードディスク…………サービス
ツクモ特価¥298,000

満開製作所の商品も取扱中！
X68000 CompactXVI 24MHz改
RED ZONE……………………ツクモ特価¥130,000
RED ZONE 2DD…………ツクモ特価¥135,000

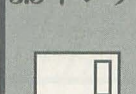
満開製外付け5インチFDD
MK-FD1………………ツクモ特価¥39,800
MK-FD1(ｶﾗｰﾘﾝｸﾞﾓﾃﾞﾙ)……ツクモ特価¥44,800
お買い得!

X68000/030シリーズ用RAMボード

品名	ツクモ特価
SH-6BE1-1ME(CZ-600C専用)	¥10,800
PIO-6BE1-AE(ACE/PRO/PRO2シリーズ用)	¥10,800
PIO-6BE2-2ME(拡張スロット用)	¥22,800
PIO-6BE4-4ME(拡張スロット用)	¥38,800
SH-5BE4-8M(X68030シリーズ用)	¥44,800
CZ-6BE2D(CompactXVI専用・RED ZONEにてOK!)	¥29,800
TS-6BE2B(CZ-6BE2A/D用拡張RAM)	¥29,800
TS-XM1-4A(拡張スロット用4MB)	¥39,800
TS-XM1-6A(拡張スロット用6MB)	¥47,800
TS-XM1-8A(拡張スロット用8MB)	¥55,800
TS-XM1-10(拡張スロット用10MB)	¥63,800
X SIMM10-8M(拡張スロット用8MB)	¥53,800
X SIMM10-10M(拡張スロット用10MB)	¥64,800

ツクモオリジナルシリーズ TS-3XRシリーズ X680x0外付ドライブ

・2DD/2HD/2HC / 1.44MBフォーマット対応
※2DD/2HC/1.44MBを使用するにはHuman68K Ver.3.0以上が必要
・CompactXVI/68030用ケーブル付

3.5インチ

TS-3XR1B 1ﾄﾞﾗｲﾌﾞ 定価¥33,800 **ツクモ特価¥26,800**
TS-3XR2B 2ﾄﾞﾗｲﾌﾞ 定価¥46,800 **ツクモ特価¥36,800**

ディスプレイも特別価格にて提供中！

CZ-607D(14型ｶﾗｰﾃﾞｨｽﾌﾟﾚｲﾃﾚﾋﾞ)……………ツクモ特価¥ 60,000
CZ-608D(14型ｶﾗｰﾃﾞｨｽﾌﾟﾚｲ)……………ツクモ特価¥ 69,000
CZ-615D(15型ｶﾗｰﾃﾞｨｽﾌﾟﾚｲﾃﾚﾋﾞ)……………ツクモ特価¥132,000
CZ-621D(21型ｶﾗｰﾃﾞｨｽﾌﾟﾚｲ)……………ツクモ特価¥125,000

ツクモオリジナル

ジョイスティック パラレルインターフェイス 近日発売予定
TS-JPIFS……定価¥17,800 **発売記念特価¥14,800**
拡張スロットを使用しません。ジョイスティック端子に接続できるパラレルインターフェイスです。これでスキャナーも高速で取り込みが可能になります。取り込みソフトェア及びサンプルソースが付属致します。

MIDIインターフェース CZ-6BM1A **ツクモ特価¥19,000**

ハードディスク

大容量記憶装置
SCSIボードが必要な場合にはセット価格に¥22,000加算となります。

120MBハードディスク……………ツクモ特価¥39,800～
200MBハードディスク……………ツクモ特価¥42,800～
240MBハードディスク……………ツクモ特価¥46,800～
340MBハードディスク……………ツクモ特価¥68,000～
540MBハードディスク……………ツクモ特価¥99,800～

プリンター

カラーバブルジェットプリンター
BJC-600J(ｹｰﾌﾞﾙｾｯﾄ)
ツクモ特価¥98,800

48ドットカラー熱転写プリンター
CZ-8PC5-BK **ツクモ特価¥38,000**

バブルジェットプリンター
BJ-10VLite(ｹｰﾌﾞﾙｾｯﾄ) **ツクモ特価¥38,500**

カラーイメージスキャナー CZ-8NS1 **ツクモ特価¥69,800**

「コレが欲しい!」とお決まりになったら、今すぐ!!
お電話一本!お気軽にどうぞ
受·注·専·用 フリーダイヤル **0120-377-999**
通販センター・・・03-3251-9911 商品についてのお問い合わせは各店または通販へ。

クレジット払い
月々¥3,000以上の均等払いも頭金なし。夏·冬ボーナス2回払いも受付中!

各種リース払い
くわしくは各店にお問い合わせ下さい。ケースに合わせてご相談承ります。

現金書留払い
〒101-91 東京都千代田区神田郵便局私書箱135号 ツクモ通販センター Oh!X係

カード払い（¥5,000以上）
通信販売でのご利用カード、ツクモグローバルカード、セントラル、ジャックス※ご本人様より電話で通販部へお申し込み下さい。

全国代金引換え配達
お申し込みは TEL03-3251-9911へお電話1本!配達日の指定もできます。

銀行振込払い
事前にTELでお届け先をご連絡下さい。三和銀行 秋葉原支店（普）1009939 ﾂｸﾓﾃﾞﾝｷ

12回払い、7.5%がナント6%に!

クレジット金利がこんなにお安くなりました! ～月々ムリのないお支払い額で欲しかったパソコンがお手元に!!～

支払回数(回)	1	3	6	10	12	15	18	20	24	30	36	42	48	54	60
分割払い手数料率(%)	2.5	3.5	4.5	5.5	6	9	11.0	12	12.5	16.5	17.5	22	23	28.5	29.5

ツクモグローバルJCBカード登場!!
好評 入会者受付中!
18才以上なら学生でもOK!
JCBならではの国内、海外サービスにツクモオリジナルの特典をプラス。お支払いはプランに合わせて、1回·2回·ボーナス一括·リボルビング払いから選べるのでとても便利!!ツクモ各店備え付けの入会申込書にてお申し込み下さい。詳しくはグローバル事務局03-3251-9898または各店へ。
★ジャックス/VISA/セントラル/マスターも取り扱っております。

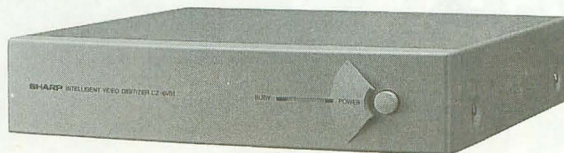
動画を始めてみませんか？

ビデオ入力ユニット CZ-6VS1

標準価格￥178,000

MC68EC020(25MHz)の32BitMPUを搭載し、SCSIインターフェイスを介してパソコンへデータを転送。動画・静止画を簡単に保存出来るアプリケーションソフト「ライブスキャン」を標準装備。1,677万色まで対応し、最大640×480ドットの高解像度で、高速取り込が可能です。但し、X680x0シリーズでご使用の場合には6万5千色までの表示となります。

ツクモ特価販売中!!

超低金利！夏・冬ボーナス二回払い受付中!! 詳しくは各店までお問い合わせ下さい。

MIDIコンピュータミュージック特選セット

Roland SC-55mkIIセット

SC-55mkII	¥69,000
SX-68MII	¥19,800
Mu-1GS	¥28,000
合計定価	¥116,800

ツクモ特価￥92,000

Roland SC-88セット

SC-88	¥89.800
SX-68MII	¥19,800
Mu-1GS	¥28,000
合計定価	¥137,600

ツクモ特価￥110,000

簡単コンピュータミュージック

Music Card for X680x0 (TS-6GM1) **発売記念価格¥39,800**

・音源を搭載したMIDIボードの登場。これ1枚で手軽にMIDIコンピュータミュージックが楽しめます。
GM規格・MT－32・CM－64等の音色配列をサポート。最大同時発音数16。

コンピュータアート

スーパーグラフィックツール NEW セット

その1.慣れてしまうとマウスがいらない

DrawingSlate	￥74,800
Matier Ver2.0	￥39,800
合計定価	￥114,600

ツクモ特価￥82,000

その2.ハイオリティなのにこんなに安い

BJC-600J	¥120,000
プリンターケーブル	￥ 4,800
Matier Ver2.0	￥ 39,800
合計定価	￥164,600

ツクモ特価￥125,000

MO特選セット

Logitec	LMO-FMX330	¥178,000
	MOメディア	付属
	SCSIケーブル	サービス
	ツクモ特価	**￥ 99,800**
コバル	CS-M120PX(ブラック)	¥178,000
	SCSIケーブル	サービス
	MOメディア	サービス
	ターミネータ	サービス
	ツクモ特価	**¥118,000**
SONY	RMO-S360	¥169,000
	MOメディア	付属
	SCSIケーブル	サービス
	ツクモ特価	**¥128,000**

CD－ROMドライブ(2倍速)

メーカー	型番	ツクモ特価
ELECOM	ECD-250(TOSHIBAドライブ)	¥47,800
東芝	XM-4100A(TOSHIBAドライブ)	¥47,800
Logitec	LCD-550-DV(TOSHIBAドライブ)	¥41,800
SONY	CDU-7811(SONYドライブ)	¥45,800
緑電子	CXA-301(NECドライブ)	¥36,800

6連装CD－ROMドライブ

PIONEER

DRM-602X(2倍速) **ツクモ特価¥ 69,800**

DRM-604X(4倍速) **ツクモ特価¥178,000**

CD－ROMドライバーソフト＋SCSIケーブル **ツクモ特価¥9,200**

パソコン通信

モデム

		ツクモ特価
AIWA	PV-AF144V5	¥37,000
OMRON	MD144XT10V	¥37,800
MicroCORE	MC14400FX	¥34,800
Panasonic	TO-703B	¥36,800

通信ソフト

SPS	た～みのる2	¥13,000
SHARP	Communication SX-68K	¥15,800

ソフトウェア

	ツクモ特価
OS-9/X68030 V2.4.5	¥20,000
Technical Tool Kit V.2.4.5	¥16,000
Ultra C & Professional Pack V1.1	¥36,000
X Windows V11.5	¥24,000
SX-WINDOW Ver3.1システムキット(NEW)	¥15,800
SX-WINDOWデスクアクセサリ集	¥11,800
C COMPILER Ver2.1 NEWKIT	¥35,800
Easydraw SX-68K	¥15,800
Easypaint SX-68K	¥10,200

	ツクモ特価
SOUND SX-68K	¥12,600
Communication SX-68K	¥15,800
Matier Ver2.0	¥29,800
CD-ROM Driver	¥ 4,800
SX-PhotoGallery	¥15,800
DoubleBookin'	¥12,800
EGWord SX-68K	¥47,800
SX-WINDOW開発キット	¥31,800
開発キット用ツール集	¥10,200
倉庫番リベンジSX-68K	¥ 5,440

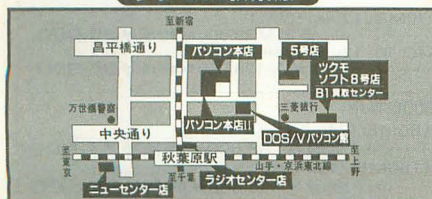
営 平日 AM10:45～PM7:30 日・祝 AM10:15～PM7:00

ツクモパソコン本店 II 4F

TEL 03(3253)1899(直通)

ツクモパソコン本店II代表 TEL 03 (3253) 4199 休 毎週木曜日

担当 荒井

ツクモIN秋葉原

ツクモニューセンター店

TEL 03 (3251) 0987

休 毎週木曜日

担当 沢栄

※下取り交換、中古販売も行っております。

ツクモIN名古屋（1号店 第一アメ横ビル内 2号店 第二アメ横ビル内）

名古屋1号店 TEL 052 (263) 1655 担当 横山 休 毎週火曜日

名古屋2号店 TEL 052 (251) 3399 担当 松原 休 毎週水曜日

ツクモIN札幌（ツクモ札幌店 DEPOツクモ2番街店）

札幌店 TEL 011 (241) 2299 担当 田口 休 毎週木曜日

DEPO店TEL 011 (242) 3199 担当 鈴木 休 毎週木曜日

各店、定休日が祝日と重なる場合は営業致します。

マイコン専門ショップ

P&A

SHARPエキスパートショップ

今が購入のチャンス!

SHARP

注目!!夏のボーナス一括払い手数料(金利)無料(平成6年6月末/7月末のいずれかをご指定下さい。)

5/18〜6/17

X68030お買い得セット

(クレジット表:送料・消費税込み)

①X68030

- CZ-500C
- CZ-607D-TN(0.31mm、チューナー付)

定価合計¥497,800
P&A超特価¥299,000

12回	27,700	24回	14,400	36回	10,000	48回	7,800	60回	6,500

②X68030 HD

- CZ-510C
- CZ-607D-TN(0.31mm、チューナー付)

定価合計¥587,800
P&A超特価¥399,000

12回	36,300	24回	19,200	36回	13,300	48回	10,400	60回	8,700

③X68030 Compact

- CZ-300C
- CZ-607D-TN(0.31mm、チューナー付)

定価合計¥487,800
P&A超特価¥329,000

12回	30,000	24回	15,800	36回	11,000	48回	8,600	60回	7,200

④X68030 Compact HD

- CZ-310C
- CZ-607D-TN(0.31mm、チューナー付)

定価合計¥577,800
P&A超特価¥394,000

12回	35,900	24回	18,900	36回	13,100	48回	10,200	60回	8,600

■モニター変更の場合
- CZ-608D……………に変更の場合¥ 3,000
- CZ-615D(チューナー付)に変更の場合¥56,000
- CZ-621D(B)…………に変更の場合¥64,000

} 加算して下さい。

X68000 Compact XVI

旧シリーズ今が買いどき!!
(クレジット表:送料・消費税込み) 送料¥2,000・消費税別

①本体+モニター

- CZ-674C-H
- CZ-608D-H

定価¥392,800
P&A超特価¥158,000

12回	14,500	24回	7,700	36回	5,300	48回	4,200	60回	3,500

②本体+モニター+FDD(5"×2)

- CZ-674C-H
- CZ-608D-H
- CZ-6FD5(FDD)

定価¥492,600
P&A超特価¥203,000

12回	18,500	24回	9,800	36回	6,800	48回	5,300	60回	4,500

③本体+モニター(TVチューナー付)

- CZ-674C-H
- CZ-607D-TN
- RGBケーブル

定価¥397,800
P&A超特価¥155,000

12回	14,200	24回	7,500	36回	5,200	48回	4,100	60回	3,400

④本体+モニター(TVチューナー付)+FDD(5"×2)

- CZ-674C-H
- CZ-607D-TN
- RGBケーブル
- CZ-6FD5(FDD)

定価¥497,600
P&A超特価¥200,000

12回	18,200	24回	9,600	36回	6,700	48回	5,200	60回	4,400

■モニターの変更 ※③、④のモニターを
- CZ-615D(チューナー付)に変更の場合¥56,000
- CZ-621D(B)　　　　　に変更の場合¥64,000

} 加算して下さい。

X68000 Compact XVI / X68000 PRO II

X68000 Compact XVI 本体(単品)	X68000 PRO II 本体(単品)	X68000 PRO II 本体+モニター
⊙CZ-674C 定価¥298,000 P&A超特価 ¥95,000	⊙CZ-653C(GY) 定価¥285,000 P&A超特価 ¥68,000	●CZ-653C(グレー) ●X68専用モニター(グレー)(14インチ) P&A超特価 ¥99,000

X68000/68030用 メモリボード

(送料¥700・消費税別)

■I/Oデータ
- SH-5BE4-8M(30用)….特価¥44,500
- SH-6BE1-1ME(600C用)…特価¥10,600
- PIO-6BE1-AE(ACE/PRO/PROII用)・特価¥10,600
- PIO-6BE2-2ME(拡張スロット用)・特価¥22,600
- PIO-6BE4-4ME(　〃　)・特価¥38,300

■シャープ
- CZ-5BE4(30用)………特価¥39,800
- CZ-5ME4(5BE4用増設)・特価¥36,500
- CZ-6BE2A(XVI用)……特価¥38,900
- CZ-6BE2B(XVI、674C増設)特価¥37,500
- CZ-6BE2D(674C用)…特価¥20,500

モデム&FAXモデム

(送料¥1,000)

通信ポコ〈インテグラン〉
- FAX MP1414F…………定価¥44,800▶特価¥31,000
- FAX MP96……………定価¥39,800▶特価¥25,000

〈アイワ〉
- PV-PF144　(FAXモデム・ポケット型)‥特価¥32,000
- PV-AF144V5(FAXモデム・ボックス型)‥特価¥38,000

〈オムロン〉
- MD-96XT 10V (FAXモデム・ボックス型)・特価¥30,000
- MD-96XL 10V (FAXモデム・ポケット型)・特価¥34,000
- MD-96FL 10V (モデム・ポケット型)……特価¥30,000
- MD-144XT10V…………………特価¥35,000

〈マイクロコア〉
- MC14400FX (FAXモデム・ボックス型)…特価¥33,000

●本広告の掲載の商品の価格については、消費税は含まれておりません。

68000 パワーアップキット(ジャスト)

(送料¥700・消費税別)

HARP+ER10Sでメモリーアクセス 約33%UP

⊙MPUアクセラレータ H.A.R.P……特価¥25,000
(600C、ACE、EXPERT、PRO、SUPER用)

⊙I/O拡張スロット ESX68L4……特価¥33,500
(4スロットに拡張、全機種対応)

⊙拡張SIMMメモリーボード ER10S 特価¥12,500
(SIMM未実装タイプ、SIMMソケット×2 全機種対応)

増設SIMM
- HT04MB-70-DV(加賀電子)・特価¥18,200
- HT08MB-70-DV(　〃　)・特価¥35,700

(例1) X68000に8M増設
ER10S+HT08MB-70-DV=¥48,200(DOS V用 72ピン、70ns)

(例2) 最大メモリ実装(10M)
ER10S+HT08MB-70-DV+HT04MB-70-DV=¥66,400

東京システムリサーチ製(X SIMM 10)

(送料¥700・消費税別)

⊙SIMM増設式メモリボード
- X SIMM 10

定価¥18,000➡特価¥15,700

増設SIMMメモリ
- 1MB×2………特価¥12,000
- 4MB×2………特価¥30,000

- 4MB例
X SIMM 10+1MB×2+1MB×2・¥39,700
- 10MB例
X SIMM 10+1MB×2+4MB×2・¥57,700

X68000/68030専用ハードディスク

(送料¥1,000・消費税別)

外付

■ロジテック
⊙SHD-B240N-FMX(ケーブル付)(240MB、14ms、64K)
……………定価¥59,800▶特価¥45,000
⊙SHD-B340N-FMX(ケーブル付)(340MB、12ms、128K)
……………定価¥74,800▶特価¥55,000

■富士通
⊙HD-K240(モッキンバード)(240MB、15ms)
……………定価¥79,800▶特価¥46,500
⊙HD-K340(モッキンバード)(340MB、12ms)‥特価¥59,800
⊙HD-K520(モッキンバード)(520MB、12ms)
……………定価¥128,000▶特価¥88,000
⊙CS-H500(Filo)(530MB、12ms)………特価¥79,000

■ジェフ
⊙GF-270(270MB、12ms、128K)
……………定価¥89,800▶特価¥59,000
⊙GF-540(540MB、12ms、128K)
……………定価¥128,000▶特価¥85,000

内蔵

■CZ-500C/300C専用
⊙CZ-5H08(80MB/23ms)
……………定価¥ 98,000▶特価¥71,800
⊙CZ-5H16(160MB/18ms)
……………定価¥135,000▶特価¥99,500

ズバリご奉仕

P&Aならではの 新品パソコン 5年保証

《業界No.1の"P&Aメンテナンスサポート"》

最高の保証システム

①業界最長の新品パソコン5年保証
（※モニター・プリンター3年間保証!! ※一部商品は除きます。）
②中古パソコンの1年間保証（※モニター・プリンター6ヶ月間保証!!）
③初期不良交換期間3ヶ月（※新品商品に限らせていただきます。）
④永久買取保証
⑤配達日の指定OK!!（土曜・日曜・祭日もOK!!）
⑥夜間配達もOK!!（※PM6:00～PM8:00の間 ※一部地域は除きます。）

便利でお得な支払いシステム

①翌月一括払い手数料無料（ご利用下さい。）
②業界No.1の低金利!!
③月々の支払いは¥1,000より
④9ヶ月先からのスキップ払いOK!!
⑤84回までの分割、ボーナス併用OK!!
⑥カレッジクレジット
⑦ステップアップクレジット
⑧ボーナスだけで10回払いOK!!
⑨現金一括支払いOK!!
⑩商品到着払いOK!!（代引き手数料が必要になります。10万円まで900円）
（※商品・金額ご確認の上、銀行振込・現金書留にてご入金下さい。）

●法人向け リースシステム
業務に最適なシステムを構築します。
損金処理が可能なリース契約をどうぞ。

周辺機器コーナー （送料¥1,000・消費税別）

カラーイメージスキャナ
■JX-325X
定価¥190,000
特価**¥99,800**

カラーイメージジェット
■IO-735X-B
定価¥248,000
特価**¥128,000**

ビデオスキャナー
■CZ-6VS1
定価¥178,000
特価**¥135,000**

FDD（5インチ×2基）
■CZ-6FD5
定価¥99,800
P&A超特価
¥49,800

プリンター（ケーブル用紙付）
- MJ-500V2 （エプソン）………特価¥48,500
- MJ-1000V2（ 〃 ）………特価¥71,300
- VP-1047PC（ 〃 ）………特価¥49,000
- BJ-220JC （キヤノン）……特価¥58,000
- BJ-10V Lite （ 〃 ）……特価¥34,500
- BJ-15V PRO （ 〃 ）……特価¥46,500
- LBP-A404GII（ 〃 ）……特価¥99,500
- BJC-820J （ 〃 ）……特価¥154,300
- JET505J PLUS（YHP）……特価¥53,300

光磁気ディスク（X68000用）
■CS-M120（コパル）
●ケーブル、ターミネータ付
¥178,000
特価**¥96,500**

- CZ-6TU…………定価¥33,100▶特価¥23,900
- CZ-8NM3………定価¥ 9,800▶特価¥ 7,200
- SH-6BF1………定価¥49,800▶特価¥36,500
- CZ-6BP1………定価¥79,800▶特価¥57,000
- CZ-6BS1………定価¥29,800▶特価¥21,500
- CZ-8NJ2………定価¥23,800▶特価¥17,500
- CZ-6CS1（674C用）・定価¥12,000▶特価¥ 8,900
- CZ-6CR1（RGBケーブル）・定価¥ 4,500▶特価¥ 3,600
- CZ6CT1（テレビコントロール）・定価¥ 5,500▶特価¥ 4,400
- CZ-6BP2………定価¥45,800▶特価¥33,300
- CZ-5MP1（X68030用）・定価¥54,800▶特価¥42,000

送料¥700・消費税別

■システムサコムボード
- SX-68MII（MIDI） 定価¥19,800 特価¥13,500
- SX-68SC（SCSI） 定価¥26,800 特価¥17,500

X68000用ソフトコーナー （送料¥700・消費税別）

- Z's STAFF PRO68K Ver.3.0（ツァイト）……定価¥58,000▶特価¥37,500
- Z's TRIPHONYデジタルクラフト（ツァイト）……定価¥39,800▶特価¥27,000
- マジックパレット（ミュージカルプラン）……定価¥19,800▶特価¥14,200
- たーみのる2（SPS）……定価¥17,800▶特価¥13,000
- Mu-1 Super（サンワード）……定価¥39,800▶特価¥28,500
- サイクロンEXPRESS α68……定価¥98,000▶特価¥69,000
- Video PC for X680X0（マイクロウェアシステムズ）……定価¥58,000▶特価¥46,400
- X WINDOWS V.11.5（マイクロウェアシステムズ）……定価¥30,000▶特価¥25,500
- Double Book IN（計測技研）……定価¥12,800▶特価¥ 9,600
- OS-9/X68030 V.2.4.5（マイクロウェアシステムズ）……定価¥25,000▶特価¥19,900
- C&Professional Pack V.3.2（マイクロウェアジャパン）……定価¥80,000▶特価¥57,800
- マチエール Ver.2.0……定価¥39,800▶特価¥28,800
- CZ-213MSD MUSIC PRO68K……定価¥18,800▶特価¥13,200
- CZ-214MSD SOUND PRO68K……定価¥15,800▶特価¥11,300
- CZ-215MSD Sampling PRO68K……定価¥17,800▶特価¥12,500
- CZ-220BSD DATA PRO68K……定価¥58,000▶特価¥40,000
- CZ-225BSV Multiword Ver.2.0……定価¥32,000▶特価¥23,000
- CZ-243BSD CYBERNOTE PRO68K……定価¥19,800▶特価¥15,000
- CZ-247MSD MUSIC PRO68K（MIDI）……定価¥28,800▶特価¥20,500
- CZ-249GSD CANVAS PRO68K……定価¥29,800▶特価¥22,000
- CZ-251BSD Hyperword……定価¥39,800▶特価¥29,400
- CZ-253BSD CARD PRO68K Ver.2.0……定価¥29,800▶特価¥22,700
- CZ-257CSD Communication PRO68K Ver.2.0……定価¥19,800▶特価¥15,300
- CZ-258BSD Teleportion PRO68K……定価¥22,800▶特価¥16,900
- CZ-261MSD MUSICstudio PRO68K Ver.2.0……定価¥28,800▶特価¥21,200
- CZ-263GWD Easypaint SX-68K……定価¥12,800▶特価¥ 9,800
- CZ-264GWD Easydraw SX-68K……定価¥19,800▶特価¥15,300
- CZ-265HSD NewPrintShop Ver.2.0……定価¥20,000▶特価¥15,400
- CZ-266BSD PressConductor PRO68K……定価¥28,800▶特価¥22,000
- CZ-267BSD CHART PRO68K……定価¥38,000▶特価¥29,800
- CZ-271BWD EG-Word……定価¥59,800▶特価¥44,900
- CZ-272CWD Communication SX68K……定価¥19,800▶特価¥14,500
- CZ-275MWD SOUND SX68K……定価¥15,800▶特価¥11,500
- CZ-284SSD OS-9/X68000 Ver.2.4……定価¥35,800▶特価¥25,600
- CZ-286BSD BUSINESS PRO68K……定価¥28,000▶特価¥20,500
- CZ-288LWD開発キット（workroom）……定価¥39,800▶特価¥29,700
- CZ-289TWD 開発キット用ツール集……定価¥12,800▶特価¥ 9,600
- CZ-290TWD SX-WINDOW ディスクアクセサリー集……定価¥14,800▶特価¥11,500
- CZ-295LSD C-Compiler PRO68K Ver.2.1 NEW KIT……定価¥44,800▶特価¥32,500

☆ゲームソフト25%OFF OK!!（一部ソフト除く）

全国通販

★頭金なし！
★即日発送

●お近くの方はお立寄り下さい。専門係員が説明いたします。
●本体単品で特価で受付します。詳しくは電話にてお問合せ下さい。
●ビジネスソフト定価の20%引きOK！TELください。

P&A特選 今月の中古特選品

- CZ-600C‥¥45,000
- CZ-601C‥¥55,000
- CZ-611C‥¥60,000
- CZ-652C‥¥65,000
- CZ-612C‥¥85,000
- CZ-603C‥¥75,000
- CZ-653C‥¥68,000
- CZ-612C‥¥80,000
- CZ-623C‥¥80,000
- CZ-674C‥¥80,000
- CZ-634C‥¥110,000
- CZ-644C‥¥155,000

※上記は単品価格、モニター別売。

新古品 限定
●CZ-674CH
●CZ-608DH
¥138,000

限定
●CZ-634CTN（チタン）（中古）
●CZ-613D（グレー）（新品）
¥190,000
（モニターをCZ-614TN（チタン）に変更の場合¥20,000加算）

新古品 限定
●CZ-644CTN
●CZ-604DB
¥228,000

中古品
●CZ-674CH
●68000専用モニター付
¥99,000

中古品
●CZ-623C-TN
●68000専用モニター
¥98,000

中古品
●CZ-644CTN
●68000専用モニター付
¥198,000

高額買取り（新品もOK）格安販売

■まずはお電話下さい。
下取り専用 買取り電話▶ ☎03-3651-1884 FAX. 03-3651-0141
■下取り・買取りで、お急ぎの方は、直接当社に来店、または宅急便にてお送りください。

買取り価格…完動品・箱／マニュアル／付属品の価格です。中古販売…1年間保証付。

●下取りの場合…価格は常に変動していますので査定額を電話で確認してください。（差額は、P&A超低金利クレジットをご利用ください。）
●買取りの場合…現品が着き次第、3日以内に高価買取金額を連絡し、振込み、又は書留でお送り致します。
●近郊の方はP&A本店に直接お持ちください。即金にて¥5,000,000までお支払い致します。

●最新の在庫情報・価格はお電話にてお問い合せください。
●買い取りのみ、または、中古品どうしの交換も致します。詳しくは電話にて、お問い合せください。
●価格は変動する場合もございますので、ご注文の際には必ず在庫をご確認ください。
●本商品の掲載の商品の価格については、消費税は、含まれておりません。
●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せください。

P&A特選 パソコンラック＆OAチェアー（消費税込み）（送料無料、離島を除く）

①3段**¥8,240** ②4段**¥9,785**

(W-640)

※全機種→キャスター付 ※フレーム色：ホワイト
※上から2番目棚板移動可能（4段）
※3段の場合、上から2番目の棚板は付いておりません。

③4段**¥12,875**（持ち帰り可能です ご来店下さい。）

※上下2分割式/スライドマウステーブル、中棚板は2段階に可動します。
※フレーム色：グレー

①**¥9,270**
●布張り ダークグレー
●ガスシリンダー

②**¥11,330**

●布張り ダークグレー
●ガスシリンダー
●肘付

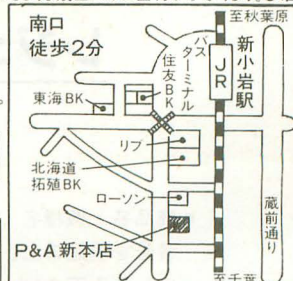
通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで現金書留でお送りください。（プリンター・フロッピーの場合、本体使用機種名を明記のこと）
〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。●1回～84回払いまで出来ます。但し、1回のお支払い額は¥1,000円以上。
〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。（電信扱いでお振込み下さい。）

〔振込先〕さくら銀行 新小岩支店
当座預金 2408626 （株）ピー・アンド・エー

超低金利クレジット率

回数	3	6	10	12	15	24	36	48	60	72
手数料	2.9	3.9	4.9	5.2	8.2	11.0	15.3	20.1	25.9	33.7

※お支払いは、便利な商品到着払い〈手数料（10万円まで900円）要〉をご利用下さい。

P&A
株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目2番地20号
●営業時間：AM10：00～PM7：00 日・祭：AM10：00～PM6：00
☎03-3651-0148（代）
FAX.03-3651-0141
●定休日／毎週水曜日

●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上お申し込み下さい。詳しくは、お電話でお問い合せ下さい。

21世紀より一歩早く

X68000が新世紀へ

レジューム・マスター"新発売"

今時のコンピュータは、レジューム機能が付いてないとね！
レジューム機能って知っているよね、ラップトップパソコンやノート型パソコンで、不精な人がそれまでのファイルをセーブしないでパソコンのスイッチを切る時に使うアレですよアレ。
レジューム機能っていうのは、スイッチを切った時にメイン・メモリー、グラフィック・メモリー、PCG、テキスト・メモリーなどをハードディスクにデータとして記録し、任意の時にそれを展開して電源スイッチを切った時の状態を再現する機能の事なんだ。
この"レジューム・マスター"は、そのレジューム機能をX68000/030で実現する常駐型ソフトなんだよ。
でも、X68000で使う場合はそれだけじゃ面白くない。
例えば、ゲームで遊んでいる時、各ステージを無傷でクリアーしたら、すかさずレジューム、そこから何回でも繰り返しPLAY、今まで難しくて最後まで出来なかったゲームが行ける様になったり。
複数のドライブにそれぞれ別々のシステムを保存してコンフィグを書き替えてリセットしなくても簡単にちがうシステムを使うことが出来ちゃう。（Z-MUSICとOPMDRVや、開発用とゲーム用のシステムを使い分ける等）
実用的には、遠くで雷が鳴っていやな雰囲気の時に、レジュームしておくと安心なんだな。（要するに停電に成りそうな時の事だよ）

だけど、こんなに便利なレジューム・マスターにもちょっとした問題も在るんだ。・・・・・・・・
FM音源の各レジスタは扱えないので、データ展開後にソフトによっては再設定しなくっちゃならない。（裏技を使えば何とかなる!?）
SX-WINDOWなどの様に自前で電源OFFルーチンを持っているソフトではレジューム機能が働かない。（そのうち対応出来るかも）
標準仕様ではない外部増設機器はサポートしていない。
S-RAMに常駐するから、S-RAMを使用するソフトとは共存する事が出来ない。（これだけはどうにも成らない）
なお、ハードディスク内に使用するコンピュータのメイン・メモリーの容量＋約1.1メガバイトの空き容量が最低必要なんだ。

★使用するコンピュータのメイン・メモリーの容量により、1～12メガバイト用の12種類有ります。
フロッピーのサイズも忘れずにお書き添え下さい。

レジューム・マスター　1～12メガバイト用
5″・3.5″2HD　各**18,000**円（消費税別）

■商品名・機種名・メディア名・住所・氏名・電話番号を明記の上、現金書留または郵便為替にてお申し込み下さい。（送料無料）

BLUE SKY Co.
株式会社　BLUE SKY
〒411 静岡県三島市加茂16-4 ☎0559-72-6710

POLYPHON

X68000 サブMPUボード ～ポリフォン～

■POLYPHONはアクセラレータではありません！

POLYPHONはサブMPUボードです。アクセラレータと異なりメイン（本体）のMPUには干渉されません。従って、メインとは別のタスクとして処理できます。ですからPOLYPHON用のアプリケーション実行させながら、別のプログラムをX68000本体で実行するといったことも可能となります。ポリフォンシステムとの組み合わせにより、DoGA（REND.X）やGCC・HAS・HLKなどの実行ファイルもX68000本体と同時に別タスクとして動作可能。POLYPHON-24使用時にはパフォーマンスが約2.0～約2.4倍に向上します。

■POLYPHONはメモリボードにもなります

POLYPHON上にはサブMPUが使用する2MBの他にX68000本体用のメモリを最大8MB搭載できます（0MB/8MBモデルとして販売）。本体用メモリ部分は純正メモリボード同様に使用できます（サブ用メモリはどちらのモデルも2MBですが、こちらは増設できません）。

■POLYPHONはコプロボードにもなります

POLYPHONはコプロを装着することが出来ます（コプロ付モデルは装着済）。コプロ部分は純正互換ですので、FLOAT3などで簡単に利用することが出来ます（コプロ機能は本体用として機能します）。コプロ無モデル購入の方は、差額にてコプロのみの販売もします。

■POLYPHONはMIDIボードにもなります

POLYPHON上にはMIDIコネクタを装備（1IN/2OUT）しています。残念ながらこちらは純正非互換ですが、Z-MUSIC,MLD,RCシステムをはじめとする各種ミュージックドライバーもPOLYPHONのMIDI OUTをサポートしているので安心です。また、市販ソフトに関してはPOLYPHON-MIDI対応パッチを用意していますので、こちらを利用すれば問題なく利用できます。（パッチはPOLYPHONシステムディスクに付属）（市販ソフトでもZ-MUSIC対応ならば、Z-MUSICの差替えのみで動作します）

■本体にない付加機能も提供します

POLYPHONには本体にない機能としてステレオPCM機能を提供しています。POLYPHON上にステレオ出力端子を備え、高品質にPCMを再生します。

POLYPHON標準価格

POLYPHON	メインメモリ8MBモデル	¥85,000-（税別）
POLYPHON	メインメモリ8MBモデル（68881付）	¥95,000-（税別）
POLYPHON	メインメモリ0MBモデル	¥62,000-（税別）
POLYPHON	メインメモリ0MBモデル（68881付）	¥72,000-（税別）

遂にアップグレード開始

大変おまたせ致しました。POLYPHON-16からPOLYPHON-24へのアップグレードが始ります。詳しいアップグレード方法はDMにてご案内致しますので、そちらをご参照下さい。

POLYPHONシステムディスクのバージョンアップを受け付けています。随時最新の内容でお届けします。ご希望のユーザーは80円切手6枚を希望メディア（3.5"または5"）を明記した上で、弊社まで送ってください。（ブランクディスク3枚と返送用切手でも可）（最新版は平成6年5月20日版）

■X680x0用外付大容量ハードディスク■

プログラム・音楽データ・画像データ…とハードディスクの足りない方にオススメ。フォーマット済のため、接続後にすぐ使用できます（パーティション分割する場合は、一旦領域解放し、再度領域を確保してください）。

・価格は定期的に変動いたしますので、最新の価格をお問い合せ願います。

お買求め・お問い合せは…

弊社製品は直販のみの販売でSHOPではお求めになれません。詳しい購入方法や細かい仕様などの最新の資料を用意しておりますので、郵便番号・住所（都道府県からお願いします）・氏名を明記の上、ハガキにてご請求ください（代金を直接送らないで下さい）。

毎日沢山の資料請求のハガキが届いておりますが、配達先不明で返送されてくるものがあります。難しい文字には読み仮名を付けて下さい。

申込みが殺到しており、発送業務が遅れており大変ご迷惑をお掛けしておりますことを深くお詫び申上げます。お申込み順に順次発送しておりますので、到着まで今しばらくお待ち下さい。

広告は今回でちょっとお休みですが、POLYPHONは継続してお申込みできます。近いうちによりパワーアップして帰ってきます。

NEO COMPUTER SYSTEMS™

株式会社ネオコンピュータシステム
120 東京都足立区綾瀬1-33-7-103
TEL 03-5680-7531（土・日・祝を除くAM10:00-PM4:00）
FAX 03-5680-7539（昨年よりFAX番号が変更になっています）
NET 03-5680-7533（サポート専用ネット）

卸店、小売店様へ

至急探しています

シャープX6800の周辺機器

カラーチューナーユニット（CZ-6TU）	カラーイメージユニット（CZ-6VT1・-BK）

この2機種について新品買取します。お問合せは下記まで。

株式会社 **アルバトロス** TEL 03-3808-0502
〒103 東京都中央区日本橋人形町3-7-6 ダイキビル4F

オリジナルアプリケーション開発速報　1994年6月1日発行（毎月1回1日発行）通巻9号　*FirstClassTechnology* 発行

X68030用　68040搭載アクセラレータ

040turbo 発進

標準価格**¥98,000**

◆040turboって、なに？

040turboは、68040を搭載したX68030用のアクセラレータです。開発者のBEEPs氏(ハンドル名)自らが、その開発過程の一部始終をまとめた書籍「X68040turbo ～A Story of Making "After X68030"～」)ソフトバンク刊(が出版されていますので、より詳しく知りたい方はぜひどうぞ。

◆私のX68030で使えるの？

040turboを装着できるのは、5インチ(130mm)FDタイプのX68030(CZ-500C/CZ-510C)です。3.5インチ(90mm)FDタイプのX68030 Compact(CZ-300C/CZ-310C)ではご使用になれませんのでご注意ください。

◆どれくらい早くなるの？

040turboで得られるパフォーマンスは、X68030の2～3倍! 計算、特に浮動小数点演算中心のソフトならば、さらにそれ以上の高速化も望めます。数値計算やレイ・トレーシングで日々X68030を酷使している人には絶対にお勧め。

◆私にも取り付けできる？

正直なところ、少し難しいです(^_^;
詳しい図入りのマニュアルが付属しますから、ある程度こういった工作に慣れた方なら楽しみながら作業していただけるはずです。
当社では取り付けサービスは行いませんので、自信のない方は工作が得意なお友達をキープしておくことをお勧めします。

◆どうすれば購入できるの？

040turboは当社のショップBASIC-HOUSEでの直販、および通販でのみお買い求めいただけます。また、大量に生産できないので、受注生産に近い形となります。ご注文いただいてからしばらくお待ちいただく場合もありますので、お早めにご注文ください。
ご注文には、下の注文書をコピーして、当社通販部まで郵送、またはFAXでお送りください。発送準備が整い次第、注文書のご連絡先へ郵便、またはお電話でお知らせいたします。お支払いはその後で。
(ご住所、電話番号は連絡のつきやすいものをご記入ください)

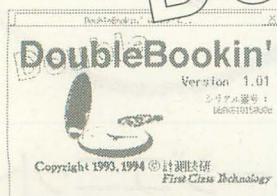
040turbo注文書

(フリガナ) お名前	郵便番号
	ご住所
数量	
ご希望の連絡方法　□郵便　□TEL.	TEL.（　　　）　-

好評発売中!　SX-WINDOW用スケジュール管理ソフト

DoubleBookin'

標準価格**¥12,800**

DoubleBookin'
Version 1.01
Copyright 1993, 1994 © 計測技研 First Class Technology

DoubleBookin'はSX-WINDOW上で動作するスケジュール管理ソフト。
長期・短期の予定の管理を支援するのはもとより、SX-WINDOWの環境に溶け込んで暮しを応援します。
たとえば…
・チェックしているドラマを見逃したくない!
・なにがなんでもOh!Xを発売日に買いたい!　(^_^)
・68+MIDI音源でゴージャスに目覚めたい!
そんなあなたに愛を込めて。DoubleBookin'がお手伝いします。
リフィル印刷や電子手帳とのリンクも万全。幅広いフィールドでお役に立ちます。

発売中　**CD-ROM Driver** Ver1.06　標準価格 ¥4,800

発売中　SX-WINDOW用Photo-CDビュアー　**SX-PhotoGallery**　基本セット ¥15,000

お求めはお近くのパソコンショップ、または弊社通販部（TEL：0286-22-9811）へお申し込みください。
通販ご希望の方は、ソフト代金＋送料￥1,000に消費税を加え、ご住所・お名前・電話番号・商品名を明記した紙を同封の上、現金封筒でお申し込みください。

※　記載されている会社名および商品名は各社の登録商標もしくは商標です。

低金利クレジット　通信販売送料 全国一律￥1,000 長期クレジット可能　※表示価格に消費税は含まれておりません

株式会社 計測技研　マイコンショップ BASIC HOUSE 本社／ショールーム／通販部　〒321 栃木県宇都宮市竹林町503-1　TEL 0286-22-9811　FAX 0286-25-3970

FreeSoftwareSelection Vol.2に多数のご応募いただきありがとうございました。現在鋭意製作中です。ご期待ください。

麻雀航海記

時は、中世ヨーロッパの大航海時代。主人公は野望に燃える貿易商。多額の遺産を元手に世界制覇を狙い、愛船の出航準備を整えていた。そこに降ってわいたようなグッドなニュース。「最も成功した若者に王女との結婚を認める。」との王様のお言葉。なんとこれぞ逆玉ではないか　とりあえず有能な参謀を使って調査したところによると、わが国との貿易を渋る国が七つ在ることが分かった。この七つの国との交易に成功した者が、晴れて王女との結婚が認められるのだ。さぁー、出航の準備にかかろう。七つの国の中には黄金の国ジパング、香辛料の国インディア・・・等があるらしい。しかも貿易の窓口は、いずれの国も若き女性らしい。あっ！そうそう、この世界では、麻雀の強い人ほど尊敬されるので、貿易の成功のためには、一にも二にも麻雀の勝敗がかかわってくるのを忘れてはいけない。それに麻雀の相手が美女ということになると・・!?。さぁもう時間が無い。食料の調達。万一のためのイカサマアイテムの調達等準備は山ほどある。「いざ！出航」

5月20日発売

TAKERU価格 ¥5,800（税込）

■対応機種／X68000,030シリーズ／要2Mバイト
■企画・制作／アレックス

ハイパーアクション／
プロレスカードゲーム

レッスルエンジェルス2
トップイベンター

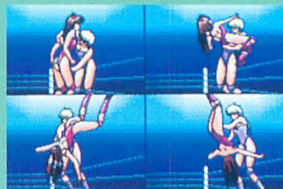
いよいよX68000登場

火花散る四角いリング!! 闘う天使達よ頂点を目指せ!!

★新人レスラーを育ててトップイベンターを目指せ！

マイティ祐希子のIWWFヘビー初戴冠より5年、日本女子マット界の勢力地図は一新された。そんな中、入門したての2人の新人、武藤めぐみと結城千種は明日のチャンピオンを目指して日夜腕をみがいていた。そして、いつしかこの2人もレスラーとしての選択を迫られていく……。

★白熱のVSモード！ 燃える対戦バトルでファイヤーしろ！

システムはおなじみのカードバトル！技が流れるマルチスポットモーションも健在だ！シングル・タッグ・アドベンチャーの各モードの他、ついにVSモードの登場で対戦プレイが可能に！もちろん恐怖の水着はぎデスマッチもパワーアップ！

TAKERU価格 ¥4,900（税込）

■X68000,030シリーズ／要2Mバイト
■企画・制作／グレイト

R.C ロボット集 +α Vol.2

ロボットコンストラクションとして大人気の「R.C.」に、またまたデータ集が登場。今回は第2回郵送ロボットバトル大会の結果発表とその参加ロボットを中心に、オンラインで行われたロボットバトル大会第3回〜5回の強豪ロボット(ベスト8)を収録。もちろんプログラムエディタや、時間別分析などのユーザー作の秀作フリーウェア、バージョンアップキットも入っています。

発売中

写真はvol.1のものです。

TAKERU価格 ¥1,800（税込）

■X68000,030シリーズ／要2Mバイト
■企画・制作／エレクトリックシープ

＊「ロボットコンストラクションR.C」が必要です。
＊「ロボット集＋α vol.1」（¥1.800）も発売中です。

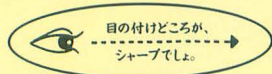
パソコンソフト自販機 TAKERU

ブラザー工業株式会社
〒467名古屋市瑞穂区苗代町2番1号

TAKERU事務局
(052)824-2493

東京営業所(03)5203−7133
大阪営業所(06)258−3024

通信販売 ■1994年4月1日より、送料/手数料が有料になりました。■

ソフト名、機種名、メディアのサイズ、住所、氏名、電話番号を明記の上TAKERU事務局まで現金書留でお申し込みください。送料／手数料は、1回のお申し込み総金額が5,000円以上の方は無料。4,900円までの方は500円をいただきます。4,900円までの方は現金500円をプラスしてお申し込みください。誠に勝手ながら、皆様のご理解とご協力の程、お願い申し上げます。（実施日'94年4月1日より）

感性を光らせる。

さまざまなフィールドで、研ぎ澄まされた感性に応える潜在能力の実証

X68の潜在能力は、まさに時代とともに証明されつつあります。
開発当初より、現在のマルチメディア環境を想定していた事実。
グラフィック能力はもちろん、ADPCM対応、オリジナルウィンドウシステム、
X68にとってこれらは、数年前のスペックなのです。
パソコンの存在そのものを革新した「創造性」、マインドを喚起する「こだわり」、
いま、先見のユーザーに支えられたX68は
そのコンセプトの開花を得て、多彩なフィールドへと飛翔します。

Workbench
WSとしての楽しみ

たとえば、リアルタイム・マルチタスク・
オペレーティング・システムOS/9。
X68030の能力を最大限に引き出す
UNIXライクな操作性と洗練された機能。
X-WINDOWや動画ツールのサポートで
さらに深い楽しみが…。

※OS/9はマイクロウェア・システムズ(株)の登録商標です。
※UNIXは、X/Openカンパニーリミテッドが独占的にライセンスする米国および他の国における登録商標です。

Create
創造するよろこび

SX-WINDOW開発支援ツールが
創造力を刺激する。
ソフト開発に必要なツールや
サンプルプログラムを多彩にバンドル、
ウィンドウ上で効率よく作業でき、
初めてプログラムに挑む人への
やさしい配慮が、創造するよろこびを
さらに高めてくれるでしょう。

Ammusement
遊びへのこだわり

X68の能力の高さを端的に示す
アミューズメントフィールド。
マインドをきわめたゲームフリークの
熱い期待に応える。
画像の美しさが感性を刺激する、
たとえばひと味違う大魔界村なら、
キミのこだわり度は今、全開!

© CAPCOM1991、1993 ALL RIGHTS RESERVED

X68030 / X68000
32bit PERSONAL WORKSTATION / PERSONAL WORKSTATION ·XVI

X68030 [本体+キーボード+マウス・トラックボール]
130㎜FD(5.25型)タイプ CZ-500C-B(チタンブラック) 標準価格398,000円(税別)・〈HD内蔵〉CZ-510C-B(チタンブラック) 標準価格488,000円(税別)

X68030 Compact [本体+キーボード+マウス]
90㎜FD(3.5型)タイプ CZ-300C-B(チタンブラック) 標準価格388,000円(税別)・〈HD内蔵〉CZ-310C-B(チタンブラック) 標準価格478,000円(税別)

X68000 XVI Compact [本体+キーボード+マウス]
90㎜FD(3.5型)タイプ CZ-674C-H(グレー) 標準価格298,000円(税別)

●ディスプレイは別売です。●消費税及び配送・設置・付帯工事費、使用済み商品の引き取り費等は、標準価格には含まれておりません。●画面はハメコミ合成です。

■お問い合わせは… シャープ株式会社 電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

T1002179060600 雑誌 02179-6